exelvision



ISSN 0769-3826 · N° 17 et 18 · 40 F

- la production en Corée
- bilan sur le R.N.I.S.
- gestion de fenêtres
- programmes de jeux

LA REVUE EXELEMENT VOTRE

17 18

Fabrication des Exeltels

Ce n'est plus un secret pour personne les constructeurs d'ordinateurs et non des moindres sous-traitent la fabrication de leurs micro-ordinateurs en Asie du sud-est (généralement la Corée), là où la main d'oeuvre est relativement bon marché Si il y a encore quelques années, les fabrications coréennes ou taiwanaises laissaient à désirer au niveau fabrication, les productions actuelles garantissent une très bonne qualité. Une qualité qui fait jaunir les japonais eux-mêmes En effet, depuis maintenant une dizaine d'années, ces pays ont investi énormément dans des usines robotisées ou simplement des usines hyper fonctionnelles où rien n'est laissé au hasard Le gage de qualité de production qui était l'exclusivité américaine, japonaise ou européenne s'est décentré au point même où l'on parle de qualité made in Taiwan ou made in Corea. Il était donc naturel que de grands constructeurs (Thomson pour la France) choisissent la Corée pour produire certains appareils de leur gamme informatique

Des prix qui garantissent des marges

Les prix sont un souci constant pour les constructeurs dans cette époque de concurrence débridée. Un bref retour en arrière (quatre ans) nous plongera dans la stupéfaction: Un ordinateur familial TI 99 4A ou Commodore 64 s'achetait 3 500 F minimum, un PC pas loin de 20 000 F Aujourd'hui, on brade du MSX à 200 F et un PC XT s'enlève à 3 000 F Jolie dégringolade Comment peuvent faire les constructeurs pour gagner de l'argent dans cette affaire ? S'ils en gagnaient au début de l'ère héroïque, ils ont commencé par en perdre progressivement jusqu'à en perdre beaucoup .Et , l'année dernière, Thomson annonçait la fermeture de son usine française responsable de la production de micro familiaux Motif manque de compétitivité. Même pas l'argument "qualité made in France" pouvait justifier le maintien de l'usine dans le groupe

Quelques chiffres

Exelvision fabriquait les EXL 100 dans les usines de la CGCT à Saint Omer Chaque machine revenait à un prix forfaitaire de X francs En choisissant de faire construire les micro-ordinateurs en Corée, cette somme s'est vue divisée par 2 C'est simple, efficace et ça laisse plus de marge en fin de compte Si une machine revenait à 100 F en production française, elle ne coûte plus que 50 F en production Coréenne Comment résister à une telle offre?

Le cocorico "made in france" s'étrangle dans cette deuxième moitié des années 80 Plus de frabrication française Est-ce à déplorer ? Chacun voit midi à sa porte Si certains pensent qu'il est dommage de ne plus construire en France, de ne plus faire travailler des français, ces derniers n'observent qu'une partie du tableau En fabriquant à l'étranger, les entreprises françaises maintiennent leur niveau de compétitivité et continuent à maintenir un équilibre économique dans toute la chaîne de distribution Sous-traitants, SAV, distribution, publicité, services Il faut admettre l'évidence En continuant de fabriquer en France, ces entreprises risquent de péricliter, voire même de s'éteindre diminuant ainsi les ressources d'autres entreprises liées à ce type de marché

Un autre point important que beaucoup ignorent En sous-traitant la fabrication matérielle, les produits conçus en France bénéficient de toutes les innovations techniques développées peu à peu pour d'autres types de matériels. Les produits deviennent plus fiables, plus faciles à fabriquer.

Exeltel made in Corea

Les Exeltels sont fabriqués en Corée par la firme Daewoo, un ténor de l'électronique coréenne. L'étoile montante de ces années 80 Troisième société d'électronique derrière la firme Gold Star et Samsung, la société Daewoo Electronics fabrique des ordinateurs, des récepteurs de télévision, des magnétoscopes, des chaînes HI-FI, etc... Daewoo fabrique des PC, produit les M06 de Thomson Daewoo dispose d'un grand savoir-faire en matière informatique Exelvision a choisi Daewoo justement à cause de ce savoir faire, mais également pour la possibilité qu'offrait cette dernière au niveau de la production de petites séries (500 pièces environ)

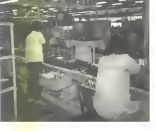
L'interview, Christian Petiot Directeur du Hard-Ware Exelvision

PC. Christian, les lecteurs aimeraient connaître les raisons pour lesquelles Exelvision fabrique ses ordinateurs en Corée. Et pourquoi Daewoo en particulier plutôt qu'une autre société ?

Fabrication des Exeltels

CP Lorsqu'en 1986 Exelvision s'est détachée de CGCT, nous ne pouvions plus bénéficier des structures de cette dernière. Il était donc impératif pour nous de trouver un groupe industriel disposant d'un savoir-faire en matière de construction informatique Notre choix s'est porté sur Daewoo car nous avions déjà travaillé avec cette société notamment pour la fourniture de moniteurs monochromes et couleurs.
De plus, il faut dire que les prix de construction que nous facturait CGCT étaient au moins deux fois supérieurs au prix du marché international. Pour une société comme la nôtre, c'était une charge trop lourde. Nous étions trop mal placés par rapport à la concurrence. Un autre point important était le niveau de qualité à peine acceptable des cartes fabriquées par CGCT


Détails de la Mass Production Line


Détails de la Mass Production Line

PC Parlons de la qualité. La production coréenne est-elle de meilleure qualité et si oui, quelle en est la cause ?

CP La qualité de production est nettement meilleure. Ceci est du plus particulièrement au sens méthodique des coréens, notamment des ouvrières coréennes placées sur les lignes de production. Ces dernières possèdent un sens inné de la précision et sont trés travailleuses. Et puis la Corée, c'est une toute autre mentalité. On parle souvent du "Womanpower" dans le miracle économique coréen. En fait, rien n'est plus exact. La qualité de production est essentiellement liée au travail de ces jeunes filles dont la moyenne d'age varie entre 18 et 24 ans et qui bénéficient d'une politique sociale au sein de l'entreprise. Les salaires de ces jeunes femmes dépassent rarement 1 400 F par mois. En France, une employée de même catégorie ne touche pas moins de 5 000 F. Une trés grosse différence à l'arrivée.

PC. Les usines coréennes sont-elles plus modernes que les usines françaises ?

CP: Pas vraiment. Non, la différence se situe au niveau du personnel employé à la production comme je l'ai dit tout à l'heure.

PC: Quelles sont les étapes de fabrication d'un micro-ordinateur ?

CP: Tout d'abord le constructeur lance une série de 5 prototypes que nous testons de manière rigoureuse. N'oublions pas que tous les composants (à part les processeurs et certains cuircuits spécialisés) sont d'origine coréenne. Il faut donc s'assurer de la qualité de ces derniers ainsi que de leur bon fonctionnement. Aprés cette série de cinq prototypes, le constructeur s'engage dans une pré-série de 300 appareils pour tester la ligne de production. Les appareils qui sortent de cette ligne sont alors testés et toute anomalie est immédiatement corrigée. Une fois cette pré-série produite, la production à proprement parler peut commencer.
Chaque carte mére (PCB ou Print Circuit Board) est positionnée en début de ligne. L'insertion automatique des composants est ensuite réalisée. Les composants axiaux tels que les résistances et

les diodes sont insérés. Viennent ensuite les composants radiaux: capacités, circuits TTL, circuits de 8, 14 ou 16 pattes. La carte ainsi equipée passe ensuite sur la PCB line où une équipe féminine place à la main les composants qui ne peuvent pas être insérés automatiquement transistors, connecteurs, fils, etc. La carte est ensuite soudée à la vague puis nettoyée dans un bain de vapeur d'eau. Les processeurs , les circuits de plus de 16 pattes , les radiateurs sont alors fixés sur la carte.
Aprés l'insertion automatique ou manuelle des composants, la carte subit une phase de test: Contrôle des infra-rouges, de la mail box, etc...



Détails de la Mass Production Line

Les cartes sont alors acheminées vers la Mass Production Line où les différentes parties du micro-ordinateur sont assemblées. Montage des transformateurs, des coques, de la face avant, pose des étiquettes, etc.. Les micro-ordinateurs sont ensuite emballés et stockés.

PC: Quelle est la cadence de production des Exeltels ?

CP: 1 Exeltel toute les 1,30 minutes

PC: Quels sont les délais liés au transport vers la France ?

CP: Les Exeltels sont expédiés par bateau et le transit prend généralement 25 jours sans compter les formalités douanières

PC: Quels sont les ennuis liés à la production d'ordinateurs à l'étranger ?

CP: Essentiellement les voyages et la vie dans les régions asiatiques. En effet, les fréquents déplacements sont un gros souci. Sans compter qu'il faut rester sur place en moyenne trois à quatre semaines pour chaque déplacement. Et puis la mentalité est vraiment différente dans ces pays. Ce que je retiendrai de la Corée, c'est le grand professionnalisme qui anime les dirigeants de ces entreprises et surtout l'art d'écouter ce que l'on a à leur dire. Quand je travaille avec des entreprises françaises, ces dernières n'écoutent qu'à moitié les conseils ou le desiderata du client. En Corée, c'est totalement différent. Ils écoutent d'abord et agissent ensuite. C'est généralement l'inverse qui se passe en France.

PC: Exelvision va-t-elle continuer à faire fabriquer ces micro-ordinateurs en Corée ?

CP: Rien ne s'oppose à ce que Daewoo reste notre fabricant. Nous sommes contents de la qualité et de la production.

PC: Y-a-t'il de nouveaux projets à Exelvision ?

CP: Oui et heureusement. Nous travaillons à l'heure actuelle sur l'intégration du clavier et du moniteur avec l'ordinateur. Tout comme le concept minitel ou Macintosh. Nous voulons produire un Exeltel compact où la connectique sera réduite de trois quarts. Et puis, effectivement, nous travaillons sur le futur terminal RNIS.

PC: L'avenir d'Exelvision pour les prochaines années ?

CP: Nous commençons vraiment à nous démarquer de nos concurrents et ce n'est un secret pour personne que nous nous orientons vers la production de terminaux télématiques sur mesure. Un marché important à l'horizon de 1992.

Le R.N.I.S

Le mot est jeté RNIS. Ce petit sigle mystérieux pour la plupart d'entre nous va bouleverser le monde des communications dans la prochaine decennie RNIS ou réseau Numérique à Intégration de Services est un fabuleux pari technologique et humain Mais qu'est-ce que le RNIS?

L'état actuel

Aujourd'hui, les télécommunications empruntent différents chemins pour transmettre les messages· Conversation téléphonique, télex, télécopie, etc. En effet, chaque type de communication passe par une ligne différente. Il faut autant de lignes que d'applications particulières. Toute la limitation du système apparaît ici L'abonné croule sous toutes les lignes installées Un vrai dédale De plus, les nouvelles formes de communication audiographie, vidéoconférence ainsi que d'autres formes de liaisons spécialisées fleurissent un peu partout Le RNIS consiste à intégrer tous ces services sur un même réseau: le réseau téléphonique numérisé commuté

Un suel réseau pour des dizaines d'applications

LE RNIS veut rassembler sur un même réseau toutes les applications qui aujourd'hui nécessitent autant de voies de transmission différentes. D'abord, un constat évident. Le téléphone reste et restera encore pendant longtemps le moyen privilégié de communication même pour les professionnels. Les raisons à cela Le réseau est relativement jeune (en France tout du moins) et dessert plus de 25 millions d'abonnés Le réseau téléphonique est en grande partie numérisé. Cela signifie simplement que la voix est échantillonnée, c'est-à-dire codée numériquement avec des zéros et des uns en opposition au réseau analogique qui transmet des variations de courant électrique. Devant l'existence du réseau téléphonique actuel, il serait stupide de se passer de celui-ci pour développer un autre réseau

Maintenant, l'objectif consiste à ne plus utiliser qu'un seul réseau pour acheminer toutes les informations Tout comme le réseau électrique qui permet indifféremment à l'abonné d'alimenter son lave-vaiselle ou même chauffer son habitation, le RNIS permettra à l'abonné de brancher sur une même prise tous les périphériques de communication téléphone intelligent, terminal, etc...

Avec le RNIS, voix, données, images, dessins, télécopies pourront voyager ensemble.
Comment cela est-il possible ?

L'augmentation des vitesses de transfert

Les vitesses de transmission des données sur un réseau analogique commuté culminent à 9 600 bits par seconde soit 1,2 Ko par seconde Avec le réseau RNIS, il est possible d'acheminer des données à une vitesse de 64 Kbits par seconde, et ce sur deux canaux plus un canal de signalisation de 16 Kbits par seconde, soit en tout 144 Kbits par seconde, 18 Ko par seconde Ces vitesses de transfert ouvrent la porte à des applications comme la télécopie rapide, l'audiovidéotex, ou la transmission d'images semi-animées pour la télésurveillance

Les différentes étapes du RNIS

Deux étapes fondamentales du RNIS ressortent à l'heure actuelle. Dans un premier temps, le RNIS ne pourra transmettre que la voix, le données, les images fixes en utilisant les canaux de transmission de 64 Kbits/s. C'est le RNIS "bande étroite" Ce système devrait fonctionner aux alentours de 1990 et empruntera le réseau téléphonique numérisé commuté actuel sans modification des lignes actuelles Dans un deuxième temps, le RNIS pourra transmettre des images animées (télévision, visiophone, programme radio). Ceci constitue le RNIS "large bande" La vitesse de tranfsert dans le RNIS à large bande avoisinera 10 Mbits/s. Le RNIS large bande ne sera possible qu'avec la pose de fibres optiques qui irrigueront les zones urbaines Ce type de RNIS fonctionnera dans le meilleur des cas vers l'horizon 2000 Pour faire fonctionner ce type de RNIS, il faudra poser des milliers de kilomètres de fibres optiques. Les fibres optiques coûtent cher et ne sont pas particuliérement faciles à poser Néanmoins, le projet de câblage existe. La seule contrainte est financière

Pratiquement, que va-t-on pouvoir faire avec le RNIS ?

RNIS bande étroite

-Parler pendant une transmission de données
-Rappel automatique d'un numéro occupé
-Identification d'un appel en instance

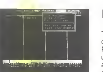






Exeltexte 80 colonnes

Traitement de texte 80 colonnes -Ecriture style gras - style normal - Insertion de textes - insertion de pages vidéotex- Définition des marges- définition des tabulations - Recherche des pages - Positionnement en début ou en de texte- Impression qualité courrier ou qualite standard
Utilisation avec imprimante type Exelwriter Exeltexte 80 ne peut être utilisé qu'avec l'Exeltel Sauvegarde sur Exelmémoire ou sur Exeldisk



Exelfile v1 ou v2

Traitement de fichiers 255 fiches de 255 caractères. Traitement des champs de saisie Recopies des champs. supression des champs Fonction de tris par ordre croissant ou decroissant ou en défissant une clé de recherche Tri multi-critères Affichage fiche à fiche ou affichage du fichier en entier Option publipostage (V1 pour exeltexte 40 colonnes V2 pour Exeltexte 80 colonnes).
Utilisation avec tout type d'imprimante Exelfile fonctionne sur EXL 100 ou sur EXELTEL Sauvegarde sur Exelmémoire ou sur Exeldisk



Cartouche Exeltel

Cartouce de mise a niveau de l'EXL 100 Cette cartouche permet de pouvoir télécharger les logiciels disponible sur le centre serveur Exelvision Cette cartouche permet notamment de télécharger les logiciels Exelquad



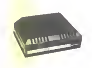
Exelmax

Exelmax est un langage d'assemblage permettant d'élaborer des programmes en langage machine Les programmes créés peuvent être ensuite intégrés dans des programmes développés en langages évolués BASIC ou LOGO Exelmax fonctionne sur EXL 100 ou sur EXELTEL Exelmax nécessite l'emploi d'une Exelmémoire ou d'un Exeldisk Livré avec manuel de programmation



Mise à niveau EXELTEL-VS

Cette mise à niveau permet de transformer un Exeltel normal en Exeltel-VS L'Exeltel-VS dispose d'un BASIC 80 colonnes et permet de programmer le port modem L'Exeltel-VS dispose en outre d'un mode de traitement des données aux normes ANSI internationales Livré avec manuel de programmation et d'utilisation La modification de l'Exeltel en Exeltel-VS est réalisée dans nos ateliers



Interface multi-fonctions

L'interface multi-fonctions se connecte à l'arrière de l'Exeltel Cette interface regroupe quatre fonctions sortie parallèle type Centronics, sortie série type RSC 232, sortie souris et incrustation vidéo Cette interface est indispensable si vous désirez connecter une imprimante. Livrée avec manuel d'utilisation et de programmation



Souris pour EXL 100 ou Exelfel

L'Exelmouse est une souris qui se connecte sur l'interface multi-fonctions L'Exelmouse est particulièrement adaptée pour l'utilisation des logiciels Exelpaint ou Exeltexte 80 colonnes Toutes les fonctions de l'Exeltel sont pilotables par la souris.





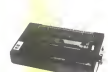
EXELDISK compatible EXL 100 /EXELTEL

Unité de disquettes 3,5 pouces D'une capacité de stockage de 640 Ko, l'Exeldisk apporte la dimension professionnelle à votre EXL 100 ou votre Exeltel Livrée avec un manuel d'utilisation et de programmation du DOS et une disquette système, l'unité de disquettes Exeldisk est indispensable pour toute application nécessitant l'emploi de fichiers L'unité de disquettes Exeldisk facilité l'emploi de votre ordinateur



Exelmémoire 16 Ko

Le périphérique indispensable L'Exelmémoire est une mémoire de 16 ko auto-alimentée permettant une sauvegarde et un chargement immédiats des données L'Exelmémoire possède un système d'exploitation de fichiers Indispensable pour le téléchargement de logiciels ou la programmation en langage d'assemblage Livrée avec manuel d'utilisation et de programmation



EXELWRITER

L'imprimante Exelwriter est particulièrement destinée au traitement de texte Exeltexte 80 colonnes L'Exelwriter est une imprimante matricielle permettant l'édition en continu ou en feuille à feuille L'Exelwriter se connecte sur l'interface multi-fonctions Silencieuse, facile d'emploi, de faibles dimensions, l'Exelwriter travaille en mode texte ou en mode graphique Impression de pages vidéotex, impression de dessin réalisés avec Exelpaint Livrée avec un manuel de programmation et d'utilisation et un ruban encreur.

Et bien sûr, tous les autres logiciels, Guppy, Wizord,Capitaine Menkar, etc - Les moniteurs monochrome ou couleur Les packs EXELTEL. Un simple coup de téléphone et on vous livre

ALLO COMMANDES
93-65-46-40

EXELDIRECT ☎ 93 65 46 40

Commandez par :

TELEX
461355 F

TELEPHONE
Appelez notre service commande au 93165146140

EXELTELEX
De 18 h à 9 h
Composez le 93165146140 avec votre Minitel ou Exeltel, au sifflement de la porteuse, appuyez sur Connexion / Fin

LE COURRIER
Remplissez le bon de commande ci-dessous et adressez le à
EXELVISION S A
Place Joseph Bermond
06560 VALBONNE

BON DE COMMANDE

LIVRAISON 24/48 H

Pour tous les envois de moins de 15 kg, vous pouvez compter sur le service 24/48 heures d'EXELDIRECT, ultra-rapide et fiable. Les frais d'expédition forfaitaires s'élèvent à 150 F.

Nom
Prénom
Adresse
Ville
Code postal Tél

Désignation des articles	Quantité	Prix TTC	Montant
		Total TTC . . .	



MODE DE PAIEMENT :
– Chèque a la commande ☐
– Contre-remboursement ☐

Nota tous nos produits sont expédiés Franco de port en cas de contre remboursement il sera facture en sus 23 70 F TTC pour expédition postale (poids inferieur à 5 kg) 58 F HT + 1 % Ad valorem pour expédition superieure a 5 kg

-Renvoi temporaire d'un appel
-Identification d'un appel
-Sélection des appels en fonction de leur provenance
-Conversation à plusieurs participants (initialement limitée à trois)
-Filtration des appels
-Utilisation d'une même prise pour tous les appareils connectés (téléfax, télex, terminal jusqu'à 8 appareils sur le bus passif)
-Possibilité de déplacer son téléphone d'une prise à l'autre sans être coupé
-Visualisation d'une image fixe sur un terminal approprié
-Télécopie à grande vitesse et en couleur
-Téléalarme
-Télécommande

RNIS Large bande

Toutes les applications décrites ci-dessus et

-Visiophonie
-Réception d'images télévisées
-Réception de programmes radio
-Téléconférences multipartites
-Interactivité de l'image
-Communication de groupe dans toutes ses applications

En fait toute la communication multimédias.

Comme vous pouvez le lire, ces applications loin d'être utopiques formeront notre environnement de demain. Les entreprises et les professionnels sont susceptibles d'être les premiers bénéficiaires da l'implantation du RNIS. Tous les métiers de la création, de la gestion, de l'édition de la prise de décision seront transformés.

Un RNIS international ?

L'implantation du RNIS à l'échelle mondiale pourrait faire appel à la transmission par satellites de télécommunication et unifierait les communications de la planète. Tout ceci est possible. Les normes du RNIS ont été définies par le CCITT (Comité Consultatif International Télégraphique et Téléphonique) Rien ne s'oppose à l'implantation d'un RNIS international. Certains pays possèdent déjà les structures pour réaliser ce réseau à moyen terme (an 2000). C'est le cas notamment de la France. D'autres nations ont un objectif qui se situe vers les années 2010 voire 2030. D'autres pays en voie de développement ne possèdent pas les infrastructures nécessaires à l'implantation d'un RNIS.

Quand fonctionnera le RNIS fançais ?

Une expérience est en cours. L'opération Renan débutera en 1988 avec trois sites des Côtes-du-Nord et de l'Ille-et-Vilaine, avant d'être étendue, à la fin de la même année, à plusieurs points de la région parisienne. Cette expérience aura pour objet de tester les nouveaux matériels mis en oeuvre ainsi que les nouveaux services proposés. A la mi-1989, Renan aura délivré tous ses renseignements et les matériels seront passés du stade de prototypes au stade de l'industrialisation.

LES FENETRES

Initiation BASIC: Gestion de fenêtres sous BASIC

Préambule

Depuis maintenant trois ans, avec l'apparition du Macintosh, une nouvelle forme d'exploitation des menus est apparue. Aujourd'hui, on parle de menus déroulants, de logiciels "user friendly". La notion de fenêtre est tellement bien implantée dans les moeurs (informatiques bien sûr) que l'on se demande même comment on a pu travailler différemment toutes ces dernières années. Pourquoi un tel succès ? L'utilisation des fenêtres facilite l'utilisation des logiciels. Tous les choix possibles sont présentés de façon claire. Bien entendu, présenter un programme sous forme de fenêtres cliquables ou saisissables demande des outils spécifiques. Le GEM de microsoft est un de ces outils. Evidemment GEM occupe une place importante dans la mémoire centrale d'un micro-ordinateur. Pour cette raison, GEM n'est réservé qu'à un certain type d'ordinateur. Tout cela est bien beau me direz-vous, mais je ne possède pas GEM sur mon EXL100 ou mon Exeltel. Dans ce numéro, nous allons développer des outils susceptibles de simuler , de générer et de gérer des fenêtres.

Concept général

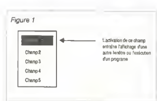
Qu'est-ce qu'une fenêtre

Une fenêtre est une partie de l'écran, donc de la mémoire contenant un certain nombre d'informations. Une fenêtre contient un ou plusieurs champs. Un champ est une zone de la fenêtre contenant une information. Un champ peut être un champ cliquable c'est-à-dire un champ qui, une fois activé à l'aide d'une touche, entraîne l'affichage d'une autre fenêtre ou l'exécution d'un programme ou une fenêtre saisissable c'est-à-dire une fenêtre qui peut recevoir une information tapée au clavier. Voir figure 1 et 2.

L'organisation des fenêtres

Si on ne tient pas compte des difficultés techniques de mise en oeuvre des fenêtres (sauvegarde du fond de l'écran), le problème de l'organisation de ces mêmes fenêtres est assez difficile à développer. En effet, chaque champ d'une fenêtre correspond à une tâche particulière

Figure 1



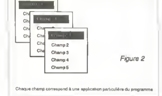


Figure 2

du programme. Par exemple, une fenêtre contenant 5 champs ou cinq rubriques (on emploiera indifféremment les deux termes) peut générer l'exécution de 5 programmes ou de cinq fenêtres. Les 5 fenêtres générées par l'activation d'un champ de la première fenêtre peuvent aussi générer d'autres fenêtres ou l'exécution d'autres programmes. Ceci est un premier point. Lorsqu'un programme est exécuté ou une fenêtre générée, il doit être possible de revenir à l'état original, c'est-à-dire avant l'activation d'un champ quelconque. Voir figure 3.

La création des fenêtres

Les problèmes de réalisation

Nous allons créer un programme de génération de fenêtres sous BASIC. Or, en BASIC, il n'est pas possible de sauvegarder le fond de l'écran. Bien évidemment, on peut créer un fichier contenant tous les paramètres de la fenêtre mais l'exécution

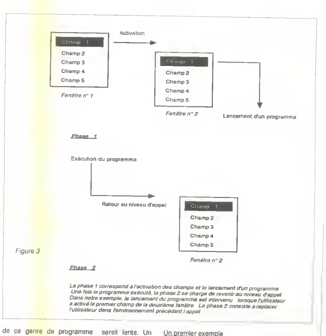


Figure 3

La phase 1 correspond à l'activation des champs et le lancement d'un programme. Une fois le programme exécuté, la phase 2 se charge de revenir au niveau d'appel. Dans notre exemple, le lancement du programme est intervenu lorsque l'utilisateur a activé le premier champ de la deuxième fenêtre. La phase 2 consiste à replacer l'utilisateur dans l'environnement précédant l'appel.

de ce genre de programme serait lente. Un programme en assembleur va nous venir en aide. Ce programme assembleur sera chargé de réserver un espace mémoire en RAM VDP (RAM VIDEO), d'afficher le cadre de la fenêtre, de sauvegarder le contenu de la fenêtre et de restituer le contenu de cette fenêtre. Chaque fois qu'une fenêtre est affichée à l'écran, son contenu est sauvegardé dans une zone du VDP.

Un premier exemple

Commençons par un exemple simple. Afficher une fenêtre sur l'écran avec un texte. Cf listing 1. Ce programme affiche une fenêtre et l'efface puis l'affiche de nouveau. Pour sortir du programme, effectuer un [CTL] [C].
Le programme BASIC appelle trois programmes assembleur qui sont successivement La

LES FENETRES

La réservation de place mémoire dans le VDP (Allocation mémoire), affichage du cadre de la fenêtre avec un passage de paramètres, la fermeture de la fenêtre.

Allocation mémoire	->	INIT.COM
Affichage	->	OPEN.COM
Fermeture	->	CLOSE.COM

Les programmes assembleur sont dans l'Exelmémoire. Pour cette raison, il faut effectuer un chargement du programme assembleur avant de le lancer à l'aide d'un CALL LOAD.

Les trois programmes assembleur

Rassurez-vous, nous n'allons pas rentrer dans des explications trop détaillées, mais néanmoins, nous examinerons la structure de ces programmes

Le programme INIT.COM

CALL DO ("INIT COM", *para1*, *para2*)

Le programme INIT COM nécessite le passsage de deux paramètres

```
                                INIT.ASM
;
;
        UNL
;       ON INCLUT LE FICHIER DES EQUIVALENCES
        COPY    C:EQU.SRC
        LIST
        BR      @DEBUT
DEBUT
        CALL    @EVALU          ;RECUPERE NOMBRE MAXIMUM DE FENETRES A GERER
        MOVD    FAC+1,TEMP8     ;RESULTAT EN ENTIER DANS TEMP8
        CLRC                    ;ON CALCULE LA TAILLE DU BUFFER DE PILE
        RLC     FAC+1           ;SOIT (NOMBRE DE FENETRES) *2 +2 OCTETS
        RLC     FAC
        ADD     %2,FAC+1
        ADC     %0,FAC
        MOVD    FAC+1,DYNLEN    ;DEMANDE D ALLOCATION D UNE ZONE MEMOIRE AU BASIC
        MOV     %MNEW,TEMP3     ;LA TAILLE DE CETTE ZONE EST DANS DYNLEN
        CALL    @BASIC          ;SI PLUS DE PLACE LA ROUTINE BASIC GENERE UNE ERREUR ET REND LA MAIN AU BASIC
        MOVD    DYNADR,TEMP9    ;SI ALLOCATION OK ADRESSE DE DEBUT DANS DYNADR
        MOVD    TEMP9,TEMP1     ;ON POSITIONNE LE POINTEUR VDP POUR ECRIRE
        TRAP    8
        CLR     A               ;PAS DE FENETRE EN COURS
        MOVP    A,WVRAM
        MOV     TEMP8,A         ;ON MET LE NOMBRE MAX DE FENETRES
        MOVP    A,WVRAM
        MOV     %GPCHR,TEMP3    ;ON FAIT AVANCER LE POINTEUR D INTERPRETATION DU BASIC POUR SAUTER LA VIRGULE
        CALL    @BASIC
        MOV     %GETADR,TEMP3   ;ON RECUPERE "L ADRESSE " DU NOM DE VARIABLE PASSE EN PARAMETRE
        CALL    @BASIC
        MOVD    TEMP9,FAC+1     ;TEMP9 CONTIENT TOUJOURS L ADRESSE ALLOUEE PAR LE BASIC
        MOV     %CIF,TEMP3      ;ON LA CONVERTIT EN FLOTTANT
        CALL    @BASIC
        MOV     %ASSGN,TEMP3    ;ON MET LA VALEUR OBTENUE DANS LA VARIABLE
        BR      @BASIC
                                ; THAT S ALL FOLKS
;                               COPIE DU SOURCE DES ROUTINES COMMUNES
        COPY    C:COMMUN.ASM
```

```
        ;       COMMUN.ASM
;
;
;APPEL AUX ROUTINES DU BASIC LE NUMERO DE LA FONCTION EST DANS TEMP3
BASIC
        PUSH    A
        PUSH    B
        MOV     TEMP3,B
        LDA     @SYSROM+1(B)    ;SYSROM EST L ADRESSE DE BASE DES POINTEURS DES
        MOV     A,TEMP3         ;ADRESSES DES FONCTIONS BASIC
        LDA     @SYSROM(B)
        MOV     A,TEMP3-1
        POP     B
        POP     A
        BR      *TEMP3          ;ON SE BRANCHE A L ADRESSE CONTENUE DANS TEMP3
;
;CETTE ROUTINES PERMET D EVALUER LES VARIABLES NUMERIQUES PASSEES SOUS BASIC
EVALU
        MOV     %GPCHR,TEMP3    ;APPEL A LA FONCTION QUI FAIT AVANCER D UN CARACTERE L INTERPRETEUR DU BASIC
                                ;ON SAUTE LA VIRGULE DE L'EXPRESSION
;                               ;CALL DO("...",var,...)
        CALL    @BASIC
        MOV     %T$DR,B         ;APPEL A LA FONCTION D EVALUATION
        MOV     %GETNUM,TEMP3
        CALL    @BASIC
        MOV     %CFI,TEMP3      ;APPEL A LA ROUTINE DE CONVERSION FLOTTANT EN ENTIER
        BR      @BASIC
;
;CALCUL D UNE ADRESSE ECRAN A PARTIR DES COORDONNEES X ET Y
ADIVDF
        MPY     %82,TEMP8-1     ;TEMP8-1 CONTIENT Y
        MOVD    B,TEMP2
        PUSH    TEMP8           ;TEMP8 CONTIENT X
        DEC     TEMP8
        RL      TEMP8
        ADD     TEMP8,TEMP2
        ADC     %0,TEMP2-1
        POP     TEMP8
        LDA     @ECRAN          ;ECRAN CONTIENT L ADRESSE DE DEBUT D ECRAN SOUS BASIC
        MOV     A,B
        LDA     @ECRAN-1
        ADD     B,TEMP2         ;EN SORTIE TEMP2 CONTIENT L ADRESSE RECHERCHEE
        ADC     A,TEMP2-1
        RETS
```

para1 . C'est un paramètre d'entrée C'est une variable numérique ou une expression numérique qui représente le nombre maximum de fenêtres.

para2 C'est un paramètre de sortie *para2* représente le nom d'une variable numérique La valeur contenue dans cette variable en retour du programme assembleur représente une adresse de stockage dans le VDP qui sera exploitée par les deux autres programmes assembleur.

Les fenêtres sont gérées comme une pile A savoir que la dernière fenêtre ouverte sera la première fenêtre qui sera fermée. Le programme INIT COM assure l'initialisation de cette gestion. Ce programme réserve en mémoire VDP une zone où sont stockés le nombre maximum de fenêtres, le nombre de fenêtres ouvertes à un instant T et les adresses de mémorisation des différentes fenêtres dans le VDP.
Le contenu de *para2* est donc l'adresse de début de cette zone dans le VDP.

Le programme OPEN.COM

CALL DO ("OPEN COM", *para2*, X,Y,ΔX, ΔY, "couleur")

Le programme OPEN.COM doit être appelé avant chaque ouverture de fenêtre.
Les paramètres du programme·

para2: C'est obligatoirement le même nom de variable que celui utilisé dans INIT.COM
X· La position en X du coin supérieur gauche de la fenêtre à l'écran entre 1 et 39.

LES FENETRES

Y La position en Y du coin supérieur gauche de la fenêtre à l'écran entre 1 et 20
ΔX La largeur en caractères de la fenêtre
ΔY La longueur en lignes de la fenêtre
"Couleur" C'est un paramètre optionnel qui peut être exprimé comme une chaîne de caractère ou contenu dans une chaîne de caractères, si l'on désire que la fenêtre soit encadrée

Exemple:

CALL DO ("OPEN.COM", para2, 1,1,39,18,"RW")

La fenêtre sera posée aux coordonnées X=1, Y=1 aura une largeur de 39 caractères et une longueur de 18 lignes, et sera affichée avec un cadre dont la couleur de fond est rouge et la couleur d'écriture est blanche

__Attention:__ Lorsqu'on écrit une fenêtre avec un cadre, ΔX et ΔY doivent en tenir compte ΔX minimum = 3 ΔY minimum = 3.

Déroulement du programme OPEN.COM.

Le programme vérifie si le nombre maximum de fenêtres permet l'ouverture d'une nouvelle fenêtre En cas de dépassement, une erreur 0,127 est générée. Si la création d'une nouvelle fenêtre est autorisée, le programme demande au BASIC d'allouer une place en mémoire suffisante pour sauver l'écran sous la fenêtre et les paramètres de la fenêtre. Si une zone mémoire est allouée, l'écran sous la fenêtre est sauvegardé dans cette zone Une mise à jour de la pile (INIT.COM) est effectuée, c'est-à-dire le compteur de fenêtre est incrémenté et l'adresse mémoire allouée par le BASIC est stockée dans l'espace mémoire de la pile

Le programme OPEN.ASM

```
        UNL
        COPY    C:EQU.SRC
        LIST
        BR      @DEBUT
DEBUT
        CALL    @EVALU
        MOVD    FAC+1,TEMP9     ;RECUPERE ADRESSE ALLOCATION DE LA PILE DE GESTION DANS TEMP9
        CALL    @EVALU          ;ON RECUPER X DANS TEMP8
        MOV     FAC+1,TEMP8
        CALL    @EVALU          ;Y DANS TEMP8-1
        MOV     FAC+1,TEMP8-1
        CALL    @EVALU          ;LAPGEUR DANS TEMP7
        MOV     FAL+1,TEMP7
        CALL    @EVALU          ;LONGUEUR DANS TEMF7-1
        MOV     FAC+1,TEMP7-1
        MOVD    TEMP9,TEMP1     ;ON POSITIONNE LE POINTEUR DE LECTURE SUR ZONE DE GESTION
        TRAP    9
        LECVDP
        MOV     A,TEMP6         ;NOMBRE DE FENETRE EN COURS
        LECVDP
        CMP     A,TEMP6         ;A CONTIENT LE NOMBRE MAX DE FENETRES
        JL      SUITE           ;SAUT SI NOMBRE DE FENETRES EN COURS INFERIEUR AU MAX
        CLR     A               ;TROP D OUVERTURES
        MOV     %127,STATUS     ;ON GENERE UNE ERREUR
        TRAP    23
SUITE                           ;CALCUL DE LA PLACE NECESSAIRE POUR SAUVER UNE FENETRE
        MOV     TEMP7,A         ;LARGEUR
        RL      A
        MPY     TEMP7-1,A       ;LONGUEUR
        ADD     %4,B
        ADC     %0,A
        MOVD    B,DYNLEN        ;TAILLE=(2*LARGEUR*LONGUEUR)+4
        MOV     %MNEW,TEMP3     ;DEMANDE D ALLOCATION D UNE ZONE DE CETTE TAILLE
        CALL    @BASIC
        MOVD    DYNADR,TEMP5    ;ON RECUPERE ADRESSE MEMOIRE POUR STOCKAGE
        CALL    @ADDTO          ;ON SAUVE ADRESSE STOCKAGE FENETRE DANS PILE GESTION
        PUSH    TEMP7-1
        CALL    @STOWIN         ;ON SAUVE LA FENETRE DANS SA ZONE
        POP     TEMP7-1
        CMP     %T$RPAR,CURCHR  ;ON VIENT TESTER SI DEMANDE D'ECRITURE D'UN CADRE
                                ;T$RPAR EST UN TOKEN DU BASIC SIGNIFIANT PARENTHSE DROITE
        JEQ     END             ;PLUS DE PARAMETRE A EVALUER
        CALL    @CADR           ;ON TRACE UN CADRE SI PARAMETRE PASSE
```

```
        MOV     %STGCLR,TEMP2           ;ON LIBERE UNE PILE INTERNE AU BASIC APRES EVALUATION
;                                        DE LA CHAINE DE CARACTERES GETSTG DANS LA ROUTINE CADR
        BR      @BASIC                  ;FIN
END     RETS
;                                       ;ROUTINE REMETTANT A JOUR LA PILE DE GESTION
ADDTO
        MOVD    TEMP9,TEMP1             ;POSITION POINTEUR LECTURE ET ECRITURE
        TRAP    9
        LECVDP                          ;NOMBRE DE FENETRES OUVERTRES
        INC     A                       ;PLUS UN CAR ON EN OUVRE UNE
        MOVP    A,WVRAM                 ;ON L'ECRIT
        DEC     A                       ;ON CALCUL L ENDROIT OU L ON DOIT ECRIRE
        MOV     A,B                     ;L ADRESSE DE SAUVEGARDE DE LA FENETRE
        CLR     A
        CLRC
        RLC     B
        RLC     A
        ADD     %2,B
        ADD     %0,A
        ADD     TEMP9,B
        ADC     TEMP9-1,A
        MOVD    B,TEMP1                 ;ON REPOSITIONNE L'ADRESSE D'ECRITURE
        TRAP    8
        MOV     TEMP5,A                 ;TEMP5 CONTIENT L'ADRESSE ALLOUEE PAR LE BASIC (LSB)
        MOVP    A,WVRAM
        MOV     TEMP5-1,A               ;TEMP5-1 EST LE MSB DE CETTE ADRESSE
        MOVP    A,WVRAM
        RETS
;                                       ;ROUTINEDE SAUVEGARDE DE LA FENTRE
STOWIN
        CALL    @ADDVDP                 ;ON CALCUL ADRESSE ECRAN DU COIN HAUT GAUCHE DE LA FENETRE
        MOVD    TEMP2,TEMP4             ;ON LE STOCK DANS TEMP4 POUR LA ROUTINE CADRE
        DECD    TEMP2                   ;ON DECREMENIE DE UN CETTE ADRESSE POUR UNE LECTURE BIDON
        MOVD    TEMP5,TEMP1             ;ON POSITONNE LE POINTEUR D ECRITURE SUR ZONE DE SAUVEGARDE DE FENETRE
        TRAP    10                      ;POINTEUR DE LECTURE (TEMP2) EST SUR L'ECRAN
        LECVDP                          ;LECTURE BIDON
        MOV     TEMP8,A                 ;ON SAUVE X
        MOVP    A,WVRAM
        MOV     TEMP8-1,A               ;ON SAUVE Y
        MOVP    A,WVRAM
        MOV     TEMP7,A                 ;ON SAUVE LARGEUR
        MOVP    A,WVRAM
        MOV     TEMP7-1,A               ;ON SAUVE LONGUEUR
        MOVP    A,WVRAM
        ADD     %4,TEMP5                ;ON AJOUTE 4 AU POINTEUR D'ECRITURE
        ADC     %0,TEMP5-1
        INC     TEMP2                   ;ON INCREMENTE LE POINTEUR DE LECTURE POUR QU'IL POINTE
        ADC     %0,TEMP2-1              SUR VALEUR CALCULEE PAR ADDVDP
        STOW10  MOV     TEMP7,TEMP3
        STOW20  LECVDP                  ;LECTUREATTRIBUT ECRAN
                MOVP    A,WVRAM         ;ECRITURE DANS ZONE SAUVEGARDE
                LECVDP                  ;LECTURE CARACTERE
                MOVP    A,WVRAM         ;ECRITURE
                DJNZ    TEMP3,STOW20    ;BOUCLE SUR LARGEUR FENETRE
                ADD     %82,TEMP2       ;POINTEUR DE LECTURE SUR LIGNE SUIVANTE
                ADC     %0,TEMP2-1
                MOV     TEMP7,A         ;ON CALCULE POSITION DU POINTEUR D'ECRITURE
                RL      A               ;CAR ON VIENT D'ECRIRE 2*LARGEUR OCTETS
                ADD     A,TEMP5
                ADC     %0,TEMP5-1
                MOVD    TEMP5,TEMP1
                TRAP    10              ;ON SE REPOSITIONNE
                DJNZ    TEMP7-1,STOW10  ;ON BOUCLE SUR LE NOMBRE DE LIGNES
                RETS
        ;                               ;SOURCE ASSEMBLEUR A INCLURE
                COPY    C:COMMUN.ASM
                COPY    C:CADRE.ASM
```

LES FENETRES

Le programme CLOSE.COM

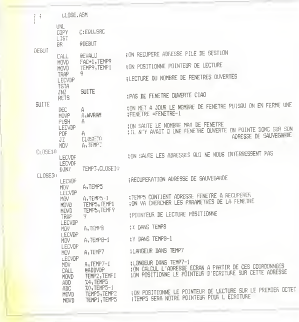
CALL DO ("CLOSE.COM", *para2*)

Le programme CLOSE.COM provoque la fermeture de la fenêtre et la restitution du fond de l'écran sauvegardé par le programme OPEN.COM.

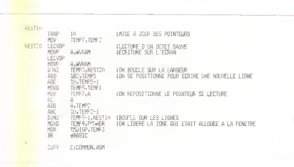
para2 C'est obligatoirement le nom de la variable numérique passée dans les deux programmes précédents

Déroulement du programme

Le programme CLOSE.COM récupère dans la mémoire de pile l'adresse de sauvegarde de la dernière fenêtre ouverte (s'il n'y a pas de fenêtre ouverte, le programme retourne au BASIC). Il récupère ensuite les paramètres de cette fenêtre, ce qui lui permet de se repositionner dans la mémoire écran. Il restitue l'écran qui avait été sauvegardé. Il libère la place mémoire allouée par le BASIC pour cette fenêtre. Il remet à jour la zone de pile



```
; ;       CLOSE.ASM
;
        UNL
        COPY    C:EQU.SRC
        LIST
        BR      @DEBUT
DEBUT
        CALL    @EVALU              ;ON RECUPERE ADRESSE PILE DE GESTION
        MOVD    FAC+1,TEMP9
        MOVD    TEMP9,TEMP1         ;ON POSITIONNE POINTEUR DE LECTURE
        TRAP    9
        LECVDP                      ;LECTURE DU NOMBRE DE FENETRES OUVERTES
        TSTA
        JNZ     SUITE
        RETS                        ;PAS DE FENETRE OUVERTE CIAO
SUITE
        DEC     A                   ;ON MET A JOUR LE NOMBRE DE FENETRE PUISQU ON EN FERME UNE
        MOVP    A,WVRAM             ;FENETRE =FENETRE-1
        PUSH    A
        LECVDP                      ;ON SAUTE LE NOMBRE MAX DE FENETRE
        POP     A                   ;IL N'Y AVAIT Q UNE FENETRE OUVERTE ON POINTE DONC SUR SON
        JZ      CLOSE30                                          ADRESSE DE SAUVEGARDE
        MOV     A,TEMP3
CLOSE10
        LECVDP                      ;ON SAUTE LES ADRESSES QUI NE NOUS INTERRESSENT PAS
        LECVDP
        DJNZ    TEMP3,CLOSE10
CLOSE30
        LECVDP                      ;RECUPERATION ADRESSE DE SAUVEGARDE
        MOV     A,TEMP5
        LECVDP
        MOV     A,TEMP5-1           ;TEMP5 CONTIENT ADRESSE FENETRE A RECUPERER
        MOVD    TEMP5,TEMP1         ;ON VA CHERCHER LES PARAMETRES DE LA FENETRE
        MOVD    TEMP5,TEMP9
        TRAP    9                   ;POINTEUR DE LECTURE POSITIONNE
        LECVDP
        MOV     A,TEMP8             ;X DANS TEMP8
        LECVDP
        MOV     A,TEMP8-1           ;Y DANS TEMP8-1
        LECVDP
        MOV     A,TEMP7             ;LARGEUR DANS TEMP7
        LECVDP
        MOV     A,TEMP7-1           ;LONGEUR DANS TEMP7-1
        CALL    @ADDVDP             ;ON CALCUL L'ADRESSE ECRAN A PARTIR DE CES COORDONNEES
        MOVD    TEMP2,TEMP1         ;ON POSITIONNE LE POINTEUR D'ECRITURE SUR CETTE ADRESSE
        ADD     %4,TEMP5
        ADC     %0,TEMP5-1
        MOVD    TEMP5,TEMP2         ;ON POSITIONNE LE POINTEUR DE LECTURE SUR LE PREMIER OCTET
        MOVD    TEMP1,TEMP5         ;TEMP5 SERA NOTRE POINTEUR POUR L ECRITURE
```



```
REST10
        TRAP    10              ;MISE A JOUR DES POINTEURS
        MOV     TEMP7,TEMP3
REST20  LECVDP                  ;LECTURE D'UN OCTET SAUVE
        MOVP    A,WVRAM         ;ECRITURE SUR L'ECRAN
        LECVDP
        MOVP    A,WVRAM
        DJNZ    TEMP3,REST20    ;ON BOUCLE SUR LA LARGEUR
        ADD     %B2,TEMP5       ;ON SE POSITIONNE POUR ECRIRE UNE NOUVELLE LIGNE
        ADC     %0,TEMP5-1
        MOVD    TEMP5,TEMP1
        MOV     TEMP7,A         ;ON REPOSITIONNE LE POINTEUR DE LECTURE
        RL      A
        ADD     A,TEMP2
        ADC     %0,TEMP2-1
        DJNZ    TEMP7-1,REST10  ;BOUCLE SUR LES LIGNES
        MOVD    TEMP9,DYNADR    ;ON LIBERE LA ZONE QUI ETAIT ALLOUEE A LA FENETRE
        MOV     %MDISP,TEMP3
        BR      @BASIC

        COPY    C:COMMUN.ASM
```

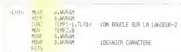
Les équivalences

Ce fichier est stocké sur Exelmémoire et s'appelle
EQU.SRC.

TEMP1	EQU	R14
TEMP2	EQU	R16
TEMP3	EQU	R18
TEMP4	EQU	R20
TEMP5	EQU	R22
TEMP6	EQU	R24
TEMP7	EQU	R26
TEMP8	EQU	R28
TEMP9	EQU	R30
TEMP10	EQU	R32
WVRAM	EQU	P46
RVRAM1	EQU	P40
RVRAM2	EQU	P36
STATUS	EQU	TEMP5
SYSROM	EQU	>0200
ECRAN	EQU	>C103
T$QR	EQU	>B0
DYNADR	EQU	R69

DYNLEN	EQU	R71
FAC	EQU	R78
ASSGN	EQU	0
CFI	EQU	>6
CIF	EQU	>0C
GETADR	EQU	>2A
GPCHR	EQU	>2C
GETNUM	EQU	>2E
MDISP	EQU	>36
MNEW	EQU	>38
STGCLR	EQU	>3C
GETSTG	EQU	>30
T$RPAR	EQU	>AF
CUCHR	EQU	R38
LECVDP	MACRO	
	PUSH	ST
	DINT	
	MOVP	RVRAM1,A

```
;       CADRE.ASM
;
;                                   ROUTINE QUI TRACE UN CADRE DANS LA FENETRE
CADR
        MOV     %GPCHR,TEMP3        ;ON FAIT AVANCER LE POINTEUR DE L'INTERPERTEUR BASIC
        CALL    @BASIC              ;POUR SAUTER LA VIRGULE
        MOV     %GETSTG,TEMP3       ;ON EVALUE LA CHAINE DE CARACTERE PASSE PAR LE BASIC
        CALL    @BASIC              ;OU LA VARIABLE DE CARACTERE
        MOVD    FAC+3,TEMP1         ;FAC+3 POINTE SUR UN DESCRIPTIF DE LA VARIABLE
        TRAP    9
        LECVDP                      ;ON LIT LA LONGUEUR DE LA CHAINE
        CMP     %2,A
        JEQ     CAD010
CAD005  MOV     %1,STATUS           ;ON N ATTEND DEUX CARACTERES ET RIEN D'AUTRE
        TRAP    23
CAD010
        LECVDP                      ;ON LIT LES CARACTERES DE LA CHAINE
        PUSH    A
        LECVDP
        MOV     %>18,TEMP5
        MOV     %8,B
CAD015  CMPA    @COL-1(B)           ;ON DECODE SI IL CORRESPOND A UN CARACTERE COULEUR BASIC
        JZ      CAD020              ;ON A TROUVE CORRESPONDANCE
        DJNZ    B,CAD015
        POP     A                   ;ET NON CE N ETAIT UNE COULEUR
        JMP     CAD005
CAD020  LDA     @BACK-1(B)          ;ON CONSTRUIT L ATTRIBUT QUI VA BIEN
        OR      A,TEMP5
        POP     A                   ;ON RECOMMENCE POUR LA COULEUR DE FOREGROUND
        MOV     %3,B
CAD025  CMPA    @COL-1(B)
        JZ      CAD030              ;OK
        DJNZ    B,CAD025
        JMP     CAD005
CAD030  LDA     @FORE-1(B)
        OR      A,TEMP5             ;TEMP5 CONTIENT L'ATTRIBUT DES COULEURS DEMANDEES
;
        MOVD    TEMP4,TEMP1         ;ON POSITIONNE LE POINTEUR D'ECRITURE SUR ECRAN
        TRAP    8
        LALL    @FLINE              ;ECRIT PREMIERE LIGNE
        SUB     %2,TEMP7-1          ;ON VA ECRIRE LARGEUR-2 LIGNES
        ADD     %82,TEMP4
        ADC     %0,TEMP4-1
        MOVD    TEMP4,TEMPI         ;ON REPOSITIONNE LE POINTEUR D ECRITURE
        TRAP    8
CAD040  CALL    @LINE
        ADD     %82,TEMP4
        ADC     %0,TEMP4-1
        MOVD    TEMP4,TEMP1
        TRAP    8
        DJNZ    TEMP7-1,CAD040      ;ON BOUCLE SUR LE NOMBRE DE LIGNES
        BR      @LLINE              ;ON ECRIT LA DERNIERE LIGNE
;                                   ;LES TROIS ROUTINES DE LIGNES NE DIFFERENT QUE PAR LES CARACTERES A ECR
FLINE
        MOVD    %>0906,TEMP2        ;>09 CARACTERE HAUT GAUCHE >06 HAUT DROIT
        MOV     %4,TEMP1            ;>04 TRAIT HORIZONTAL
        JMP     TLINE
;
LLINE   MOVD    %>0807,TEMP2        ;>08 BAS GAUCHE >07 BAS DROIT
        MOV     %4,TEMPI            ;>04 TRAIT HORIZONTAL
        JMP     TLINE
;
LINE    MOVD    %>0303,TEMP2        ;>03 TRAIT VERTICAL
        MOV     %0,TEMPI            ;>0 CARACTERE BLANC
TLINE   MOV     TEMP2-I,B
        MOV     TEMP5,A             ;TEMP5 CONTIENT ATTRIBUT VDP
        MOVP    A,WVRAM             ;ECRITURE ATTRIBUT
        MOVP    B,WVRAM             ;ECRITURE PREMIER CARACTERE
        MOV     TEMP7,TEMP1-I
        SUB     %2,TEMP1-I
        MOV     TEMP1,B
```



```
TL010   MOVP    A,WVRAM
        MOVF    B,WVRAM
        DJNZ    TEMP1-1,TL010   ;ON BOUCLE SUR LA LARGEUR-2
        MOV     TEMP2,B
        MOVF    A,WVRAM
        MOVP    B,WVRAM         ;DERNIER CARACTERE
        RETS
;                                TABLES D'EQUIVALENCE COULEUR ET DE CONSTRUCTION D'ATTRIBUT
COL     TEXT     BWbGCMYR'
BACK    BYTE    0,7,4,2,6,5,7,1
FORE    BYTE    0, E0, 80, 40,>C0, A0
        BYTE    >60, 20
```

COMMUN.ASM

Ce source contient des routines communes à INIT.ASM, OPEN.ASM et CLOSE.ASM. Si vous ne possédez pas le langage Exelmax, rapportez-vous à la fin de cette section pour réaliser l'implantation de ces routines sous BASIC

Le programme BASIC

Ligne 150	Chargement du programme d'initialisation
Ligne 160	Exécution du programme d'initialisation avec passage des paramètres: 1 fenêtre et ADX
Ligne 170	Chargement du programme d'ouverture et d'affichage de la fenêtre
Ligne 180	Exécution du pogramme d'ouverture et d'affichage avec passage des paramètres: ADX, coordonnées de la fenêtre X=1, Y=1, longueur de la fenêtre L=39, hauteur de la fenêtre H=20, et la couleur d'affichage de la fenêtre
Lignes 190 à 220	Affichage d'un texte dans la fenêtre
Ligne 250	Chargement du programme de fermeture de la fenêtre
Ligne 260	Exécution après la frappe d'une touche du programme de fermeture de la fenêtre avec un passage de paramètres

```
100 '#################################
110 '#   PARAMETRAGE FENETRE         #
120 '#################################
130 CLS "OBP":CALL COLOR "OGB")
140 LOCATE 0,0 :PRINT RPT$(" ",40)
150 NMAX=
    CALL LOAD("INIT.COM")
160 CALL DO("INIT.COM",NMAX,ADX)
    CALL LOAD("OPEN.COM")
180 CALL DO("OPEN.COM",ADX,1,1,39,20,"Wo")
190 ...
200 ... PRINT 'FFRRODGGRRAAMMMMEE'
210 CALL ...
220 LOCATE ... :PRINT "Appuyez sur une touche"
230 CALL ...  "9bFF"
240 R$=KEY$
250 CALL LOAD("CLOSE.COM")
260 CALL DO("CLOSE.COM",ADX)
```

Listing 1

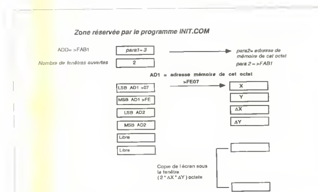
Ce programme ouvre une fenêtre, affiche un texte à l'intérieur de celle-ci et efface la fenêtre après la frappe d'une touche quelconque du clavier

Ce programme est figé dans le sens où il affiche une fenêtre dans un secteur de l'écran déterminé à l'avance. Rien ne nous empêche d'introduire les coordonnées, le texte, et la couleur d'affichage de la fenêtre durant l'exécution du programme. Ceci permet de poser une fenêtre à un endroit quelconque de l'écran.

Listing 2

Ce programme est facile à réaliser. Il faut simplement introduire les différents paramètres nécessaires au tracé de la fenêtre et le texte des rubriques

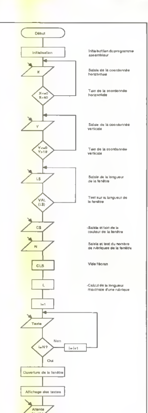
LES FENETRES



Cet exemple est pris pour trois fenêtres maximum, et deux fenêtres ouvertes

Listing 2

```
100 '################################
110 '#   PARAMETRAGE FENETRE        #
120 '################################
130 CALL LOAD("INIT.COM")
140 CALL DO("INIT.COM",1,ADX1)
150 CLS "GBB":CALL COLOR("0GB")
160 '################################
170 '#    SAISIE DES PARAMETRES     #
180 '################################
190 LOCATE (5,5):PRINT "Position en x:";
200 ACCEPT BEEP VALIDATE(DIGIT)NULL(1),X
210 IF X =0 THEN 190:IF X 39 THEN 190
220 LOCATE (6,5):PRINT "Position en y:";
230 ACCEPT BEEP VALIDATE(DIGIT)NULL(1),Y
240 IF Y =0 THEN 220:IF Y 19 THEN 220
250 LOCATE (7,5):PRINT "Longueur:";
260 ACCEPT BEEP VALIDATE(DIGIT)SIZE(2)NULL(5),L$
270 IF VAL(L$) 40-X THEN 250
280 LOCATE (8,5):PRINT "Couleur:";
290 ACCEPT BEEP VALIDATE("BbWYGRMC")SIZE(1)NULL("W"),C$
300 LOCATE (9,5):PRINT "Nombre de rubriques: ";
310 ACCEPT BEEP VALIDATE("12345")SIZE(1)NULL("1"),N
320 CLS
330 L=VAL(L$)-2
340 FOR I=1 TO N
350 LOCATE (5,5):PRINT "Texte:";
360 ACCEPT BEEP VALIDATE(ALPHA)SIZE(L)NULL("A"),T$(I)
370 NEXT I
380 FOR X=1 TO 20
390 LOCATE (X,1):PRINT RPT$(" ",40)
400 NEXT
410 CALL LOAD("OPEN.COM")
420 CALL COLOR("0"&C$&"B")
430 CALL DO("OPEN.COM",ADX1,Y,X,L+2,N+2,C$&"B")
440 FOR I=1 TO N
450 LOCATE (I+Y,X+1):PRINT T$(I)
460 NEXT I
470 CALL COLOR("0"&C$&"I")
480 I=1
490 LOCATE (I+Y,X+1):PRINT T$(1)
500 CALL LOAD("CLOSE.COM")
510 RP$=KEY$
520 CALL DO("CLOSE.COM",ADX1)
530 PAUSE
540 GOTO 410
```



Ordinogramme de principe

Les lignes du programme

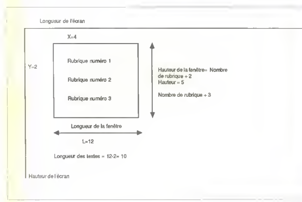
La structure de la saisie: Les lignes de programme faisant intervenir une saisie doivent prendre en compte les différentes limites physiques de l'écran: Position en X et en Y qui ne doivent en aucun cas déborder, c'est-à-dire une coordonnée horizontale X comprise entre 1 et 40, une coordonnée verticale Y comprise entre 1 et 20. De plus, la longueur et la hauteur de la fenêtre doivent répondre à certains critères. Longueur de la fenêtre inférieure au nombre de colonnes (40) diminué de la position de départ (X) L<40-X. La hauteur de la fenêtre sera considérée comme le nombre de rubriques augmentée de 2. Tous ces paramètres donnent une surface de fenêtre capable: 1- d'être affichée à l'écran sans erreur de débordement, 2- de contenir toutes les rubriques et leur texte en hauteur et en longueur. La couleur de la fenêtre est ensuite saisie dans un choix possible de 8 couleurs.

Une fois ces limites physiques définies, le programme commence la saisie des textes des rubriques. Ces textes doivent être limités à la longueur de la fenêtre diminuée de 2 unités (ceci permet d'occuper la totalité de l'espace libre de la fenêtre sans empiéter sur le tracé du cadre).

Lorsque la saisie des textes est terminée, on remplit l'écran de signes (lignes 380 à 400). Ceci permet de visualiser après l'effacement de la fenêtre le bon fonctionnement du programme assembleur chargé de mémoriser et d'afficher le fond graphique occulté durant l'affichage de la fenêtre.

Le programme chargé d'ouvrir la fenêtre est dans un premier temps chargé (ligne 410) et ensuite exécuté (ligne 430). Les paramètres de définition de la fenêtre sont transmis dans ce programme (ADX1, X, Y, L+2, N+2, et C$). La fenêtre s'affiche. Les textes des rubriques sont ensuite positionnés correctement puis écrits à l'intérieur de la fenêtre (lignes 440 à 460). La première rubrique doit être affichée en mode vidéo inverse. Ceci est réalisé aux lignes 480 à 490.

Les lignes 500 à 530 sont chargées d'attendre la frappe d'une touche du clavier puis de fermer la fenêtre et de restituer le fond sous la fenêtre. Le programme dédié à la fermeture de la fenêtre est dans un premier temps chargé (ligne 500) puis



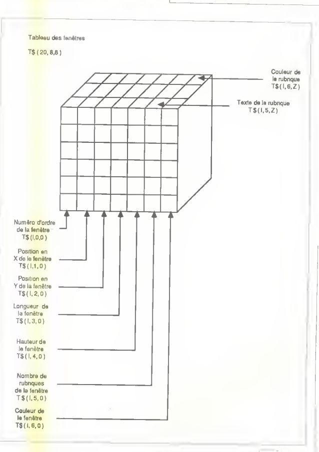
exécuté (ligne 520) avec la transmission d'un paramètre
Le programme boucle sur l'affichage et la fermeture d'une fenêtre

Comment ouvrir plusieurs fenêtres à l'ecran ?

Nous savons ouvrir une fenêtre Maintenant, nous allons nous attacher à ouvrir plusieurs fenêtres à l'écran. Mais avant de nous lancer dans la programmation, réfléchissons un peu et essayons d'analyser la situation. A chaque fenêtre, correspond un nombre de paramètres fixes qui sont les coordonnées hrizontale et verticale, la longueur, la hauteur, le nombre de rubriques, le texte des rubriques et la couleur de la fenêtre. Tous ces paramètres doivent être disponibles durant l'exécution du programme. Plusieurs possibilités s'offrent à nous. Les paramètres peuvent être écrits sous forme de données dans des instructions DATA ou être introduits durant le déroulement du programme. La première solution correspondra au cas où le nombre de fenêtres et leur contenu sont connus d'avance. La deuxième solution peut répondre à la création spontanée de fenêtres pour générer une application sur mesure. Dans tous les cas, les paramètres seront écrits dans un tableau à N dimensions

Structure du tableau

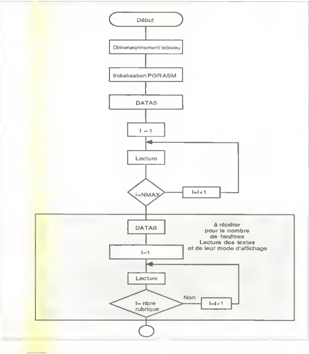
Le tableau des fenêtres que nous appellerons T$ sera un tableau à trois dimensions. La première colonne du tableau contiendra les numéros d'ordre des fenêtres. Les deuxième et troisième colonnes contiendront les coordonnées horizontale et verticale. La quatrième colonne contiendra la longueur, la cinquième colonne contiendra la hauteur de la fenêtre, la sixième colonne contiendra le nombre de rubriques, la septième colonne contiendra la couleur de la fenêtre. La sixième et la septième colonne "s'ouvriront" sur une autre dimension du tableau qui sera chargée de recevoir le texte des rubriques et leur mode d'affichage et d'autres paramètres que nous étudierons par la suite.


Tableau des fenêtres
T$ (20, 8,8)
Couleur de la rubrique T$ (I, 6, Z)
Texte de la rubrique T$ (I, 5, Z)
Numéro d'ordre de la fenêtre T$ (I,0,0)
Position en X de la fenêtre T$ (I, 1, 0)
Position en Y de la fenêtre T$ (I, 2, 0)
Longueur de la fenêtre T$ (I, 3, 0)
Hauteur de la fenêtre T$ (I, 4, 0)
Nombre de rubriques de la fenêtre T$ (I, 5, 0)
Couleur de la fenêtre T$ (I, 6, 0)

Remplissage du tableau à l'aide de DATAS

Nous nous plaçons ici dans un cas où l'application est connue d'avance et où les besoins sont parfaitement déterminés Par exemple, nous supposerons que l'application nécessite l'utilisation de trois fenêtres Une première fenêtre contiendra trois rubriques et les deux fenêtres restantes contiendront deux rubriques Le tableau général doit être intialisé. Le nombre de fenêtres maximum NMAX est configué à 3

Ordinogramme de principe



```
     '################################
     '#   PARAMETRAGE FENETRE        #
     '################################
130 DIM T$(3,8,9)
140 NMAX=3'NOMBRE DE FENETRES MAX
150 CALL LOAD("INIT.COM")
160 CALL DO("INIT.COM",NMAX,ADX1)
170 [illegible] GBG :CALL COLOR "(GB")
180 '################################
190 '#            PARAMETRES        #
200 '################################
210 DATA 1,1,1,15,5,3,W
220 DATA 2,15,10,10,4,2,M
230 DATA 3,1,1,39,4,2,Y
240 FOR I=1 TO NMAX
250 READ T$(I,0,0),T$(I,1,0),T$(I,2,0)
260 READ T$(I,3,0),T$(I,4,0),T$(I,5,0)
270 READ T$(I,6,0)
280 NEXT I
290 DATA FICHIER,1,IMPRIMER, ,SAUVER,
300 FOR I=1 TO VAL(T$(1,5,0))
310 READ T$(1,5,I),T$(1,6,I)
320 NEXT
330 DATA EDITER,1,COPIER," "
340 FOR I=1 TO VAL(T$(2,5,0))
350 READ T$(2,5,I),T$(2,6,I)
360 NEXT I
370 DATA COULEUR,1,FOND," "
380 FOR I=1 TO VAL(T$(3,5,0))
390 READ T$(3,5,I),T$(3,6,I)
400 NEXT I
410 '################################
420 '#   REMPLISSAGE DE L ECRAN     #
430 '################################
440 CLS
450 FOR I=1 TO 20
460 LOCATE (I,1):PRINT RPT$("_",40)
470 NEXT
480 '################################
490 '#     EXPLOITATION             #
500 '################################
510 FOR I=1 TO NMAX
520 X=VAL(T$(I,1,0))
530 Y=VAL(T$(I,2,0))
540 L=VAL(T$(I,3,0))
550 HH=VAL(T$(I,4,0))
560 C$=T$(I,6,0)
570 '################################
580 '#    APPEL ROUTINE CADRE       #
590 '################################
600 GOSUB 780
610 GOSUB 8[illegible]
620 PAUSE

640 NEXT
650 FOR I=1 TO NMAX
660 GOSUB 720
670 NEXT
680 GOTO 450
690 '################################
700 '#    FERMETURE FENETRE         #
710 '################################
720 CALL LOAD("CLOSE.COM")
730 CALL DO("CLOSE.COM",ADX1)
740 PAUSE
750 RETURN
```

```
760 '################################
770 '#   ROUTINE CADRE              *
780 '################################
790 CALL LOAD("OPEN.COM")
800 C$=C$&"B
810 CALL DO("OPEN.COM",ADX1,X,Y,L,HH,C$)
820 RETURN
830 '################################
840 '#    ECRITURE DES TEXTES       #
850 '################################
860 RUB=VAL(T$(I,5,0)):X=X+1
870 FOR J=1 TO RUB
880 Y=Y+1
890 LOCATE (Y,X):COUL$=T$(I,6,J):C$=SEG$(C$,1,1)[illegible]
900 X$=SEG$(C$,2,1):IF X$=" " THEN C$=SEG$(C$,1,1)
910 CALL COLOR("#"&C$)
920 LOCATE (Y,X):PRINT T$(I,5,J)
930 NEXT
940 RETURN
```

Les lignes du programmes

Le programme commence par le dimensionnement du tableau des fenêtres, toujours prévoir plus de mémoire Le programme d'initialisation des fenêtres est chargé puis exécuté

Lignes 210 à 230
Ecriture d'un jeu de données (paramètres physiques des fenêtres)

Lignes 240 à 280
Lecture et rangement des paramètres dans les 6 premières cases du tableau

Ligne 290
Ecriture d'un jeu de données correspondant aux textes des rubriques et à leur mode d'affichage de la première fenêtre

Lignes 300 à 320
Lecture et rangement dans la troisième dimension du tableau

Ligne 330
Ecriture d'un jeu de données correspondant aux textes des rubriques et à leur mode d'affichage de la deuxième fenêtre

Lignes 340 à 360
Lecture et rangement dans la troisième dimension du tableau

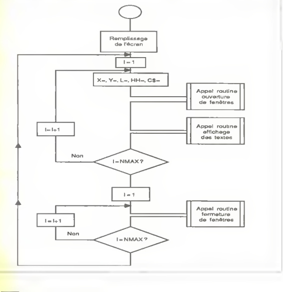
Ligne 370
Ecriture d'un jeu de données correspondant aux textes des rubriques et leur mode d'affichage de la troisième fenêtre

Lignes 380 à 400
Lecture et rangement dans la troisième dimension du tableau

LES FENETRES

Lignes 410 à 470
Remplissage de l'écran avec des soulignés (ceci permettra par la suite de test pour le bon fonctionnement du programme assembleur chargé de restituer le fond de l'écran occulté par une fenêtre)
Lignes 480 à 560
Calcul des paramètres physiques d'uns fenêtre
Ligne 600
Appel de la routine assembleur chargée de tracer le cadre d'une fenêtre avec les paramètres calculés aux lignes précédentes
Ligne 610
Appel du sous-programme BASIC chargé d'afficher les textes des rubriques pour une fenêtre donnée
Ligne 620
Temporisation, attente de la frappe de la touche [**ENVOI**]
Lignes 720 à 730
Appel à la routine assembleur chargée de fermer une fenêtre donnée. Les fenêtres sont fermées dans l'ordre inverse de leur apparition à l'écran (3ème, 2ème puis 1ére)

Ce programme affiche donc trois fenêtres et leur contenu à l'écran puis les efface. Lorsque toutes les fenêtres sont effacées, elles réapparaissent. Pour sortir du programme, appuyez sur **[CTL] C.**



Maintenant, nous allons créer un nombre quelconque de fenêtres. Le nombre, les positions, la couleur, les textes seront introduits pendant l'exécution du programme.

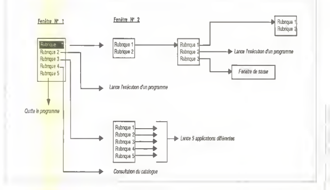
Voir listing définitif

Le programme présenté permet de créer des fenêtres, de les positionner sur l'écran, de définir le texte des rubriques

Les fenêtres et les arborescences

Nous savons créer des fenêtres. Cela n'est pas tout. Il faut maintenant pouvoir exploiter les fenêtres. En effet, chaque rubrique de la fenêtre correspond à une tâche bien particulière que le programme devra effectuer ou correspond à une fenêtre qui elle même génèrera l'ouverture d'autres fenêtres. Bien entendu, il faut avoir une vision précise de l'application

Schéma



Effectivement, on risque de s'y perdre parmi ce dédale de fenêtres et d'applications. De quoi avons-nous besoin pour arriver à ce résultat? Chaque fenêtre et a fortiori, chaque rubrique est susceptible soit de générer le lancement d'une application (cas de la rubrique n°2 de la fenêtre n° 1) soit de générer l'ouverture d'une autre fenêtre (cas de la rubrique n°1 de la fenêtre n°1). Chaque rubrique d'une fenêtre peut être sélectionnée (déplacement du mode vidéo inverse d'une rubrique à l'autre) et activée par l'appui sur la touche [**ENVOI**] Ceci implique la programmation d'une routine chargée de déplacer le mode vidéo inverse de proche en proche et d'une autre routine capable d'activer une rubrique ou un champ. De plus une fenêtre et ipso facto une rubrique peut être désactivée (généralement par l'appui sur la touche [**ESC**] pour revenir au niveau supérieur. C'est-à-dire que si la rubrique n°1 ou 2 de la fenêtre n°2 est désactivée, le programme doit afficher le niveau supérieur (niveau avant l'activation et l'affichage de la deuxième fenêtre) qui est ici la fenêtre n°1. De même, si la rubrique n°2 de la fenêtre numéro 1 est activée, une application est lancée. Lorsque l'application est terminée, le programme doit revenir au niveau supérieur, qui dans ce cas est la fenêtre n°1. Joli casse-tête en perspective, mais,

ne nous décourageons pas, la suite est encore plus compliquée.

De ce fait, chaque rubrique doit être accompagnée de son "plan d'action". Expliquons-nous. Prenons par exemple la rubrique n°1 de la fenêtre n° 1. On connaît son niveau d'activation. C'est l'affichage de la fenêtre numéro 2. Toute l'astuce consiste à renseigner le programme sur ce niveau. Ce n'est pas véritablement un problème, car nous utilisons un tableau à trois dimensions qui peut, outre le fait de contenir le texte des rubriques, intégrer de nombreux autres renseignements.

Fenêtre n°1

	Départ	Arrivée
Rubrique n°1	1	2
Rubrique n° 2	1	PGR**
Rubrique n° 3	1	3
Rubrique n° 4	1	PGR
Rubrique n° 5	1	0*

Fenêtre n° 2

	Départ	Arrivée
Rubrique n°1	2	4

Fenêtre n°3

	Départ	Arrivée
Rubrique n°1	4	5
Rubrique n° 2	4	PGR
Rubrique n° 3	4	5

* 0 indique l'abandon du programme en cours

**PGR indique le lancement d'un programme

Niveau de départ et niveau d'arrivée

On définit un niveau de départ comme l'état de la fenêtre avant l'activation. Si nous prenons la fenêtre n°1, le niveau de départ est 1. En effet, il n'existe aucune autre fenêtre antérieure. On définit le niveau d'arrivée comme l'état de la fenêtre après l'activation d'une rubrique. Par exemple, pour la rubrique n° 1 de la fenêtre n° 1, le niveau d'arrivée est 2.

Cela signifie que si la rubrique n° 1 de la fenêtre n° 1 est activée, le programme affiche la fenêtre n°2. En fait dans ce cas, le niveau d'arrivée de la fenêtre n°1 devient le niveau de départ de la fenêtre n° 2. Reprenons le schéma précédent et écrivons les niveaux de départ et les niveaux d'arrivée des différentes fenêtres. Nous notons que pour les rubriques 2 et 4 de la fenêtre n° 1, le niveau d'arrivée est un programme.

Adaptation des niveaux au programme

Il faut maintenant adapter les niveaux de départ et d'arrivée au programme. Ceci peut être réalisé en ajoutant quelques lignes d'instructions dans la saisie des paramètres physiques de la fenêtre. Ces niveaux seront écrits et rangés dans des



éléments de la troisième dimension du tableau des fenêtres lors de la saisie des rubriques Pour réaliser cette opération, il suffit d'écrire ce groupe de lignes (Cf listing final)

```
600 LOCATE (5,5): PRINT "Texte:";
610 ACCEPT BEEP VALIDATE ( ALPHA )
SIZE (L ) NULL ("A"), T$( I, 5, J )
620 T$(I,1, J )=STR$(I)
630LOCATE (6,5 ) : PRINT "Niveau de
départ:";
640PRINT I
650 LOCATE ( 7,5 ): PRINT "Niveau
d'arrivée:";
660 ACCEPT BEEP SIZE (17 ) NULL ( 0 ),
T$( I,2, J )
```

Comme nous l'avons écrit plus haut, le niveau d'arrivée d'une rubrique peut être un programme A ce sujet, nous devons savoir qu'il est pratiquement impossible de lancer une application contenue dans le programme principal Si le niveau d'arrivée de la rubrique est un programme, ce dernier devra se situer sur l'Exelmémoire ou sur disquette Le niveau d'arrivée se résume alors au nom du programme Nous verrons par la suite comment lancer le programme

Déplacement du mode vidéo inverse dans les fenêtres

Pour sélectionner une rubrique au sein d'une fenêtre, nous devons simuler le déplacement du "curseur" dans cette fenêtre Pour réaliser ce petit bout de programme, nous devons obligatoirement partir d'un point de départ Ce point de départ est l'affichage de la fenêtre 1 avec la rubrique 1 affichée en mode vidéo inverse
Les trois programmes que nous avons écrits affichent toutes les fenêtres sans discrimination Ceci ne nous intéresse plus à présent Bien au contraire Une fois la saisie des paramètres fenêtres réalisée, le programme doit afficher la première fenêtre et attendre la frappe des touches de commandes Ces touches de commandes sont la touche [**ENVOI**] pour l'activation d'un champ ou d'une rubrique, la touche [**ESC**] pour la désactivation d'un champ ou d'une rubrique, la touche [**Fleche vers le bas**] pour la sélection d'une rubrique ou d'un champ

Affichage de la fenêtre n°1

Lorsque tous les paramètres des fenêtres sont introduits, le programme affiche la fenêtre n°1. Ceci se réalise très simplement (voir listing 5) Après la saisie des textes des rubriques et des différents niveaux, nous avons rempli d'autres éléments de la troisième dimension du tableau avec les coordonnées des différents textes à afficher

```
730  T$(I,3,J) = STR$ (X+1)
740 T$(I,4,J) = STR$ (Y+J)
```

La ligne 750 fixe le mode d'affichage de chaque rubrique. Le mode d'affichage de la première rubrique de chaque fenêtre est fixé au mode vidéo inverse, ligne 790

Grâce aux lignes 730, 740, 750 et 790, nous avons tous les éléments pour déplacer de proche en proche le mode vidéo inverse L'affichage à proprement parler de la première fenêtre que vous aurez définie, se réalise simplement en donnant I = 1 et en convertissant les chaînes de caractères contenues dans les élements du tableau Pour mémoire:

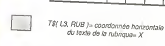
T$(I,1,0) = X	Coordonnée horizontale
T$(I,2,0) = Y	Coordonnée verticale
T$(I,3,0) = L	Longueur de la fenêtre
T$(I,4,0) = HH	Hauteur de la fenêtre
T$(I,1,0) = C$	Couleur de la fenêtre

La saisie et le traitement des touches

La saisie et le traitement des touches de commandes (déplacement du mode vidéo, activation et désactivation) sont traités aux lignes 800 à 860 A la sortie de ce bout de programme, le code de la touche appuyée peut prendre les valeurs 1, 2, 3, 4

Code	Action
1	Activation
2	Déplacement vers le bas
3	Désactivation (retour au niveau supérieur)
4	Déplacement vers le haut

La partie grisée représente la troisième dimension du tableau. Représentons cette troisième dimension

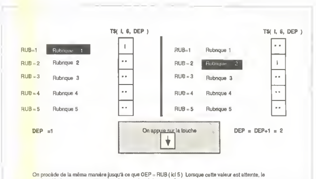


T$(I,3, RUB)= coordonnée horizontale du texte de la rubrique= X

T$(I,4, RUB) = coordonnée verticale du texte de la rubrique = Y

T$(I, 5, RUB)= Texte de la rubrique

T$ (I, 6, RUB) = mode d'affichage



On procède de la même manière jusqu'à ce que DEP = RUB (ici 5). Lorsque cette valeur est atteinte, le mode vidéo inverse est remis en fonction pour la rubrique numéro 1

Une instruction de branchement à quatre sous-programmes (ligne 1140) conclut le traitement.

Le déplacement vers le bas

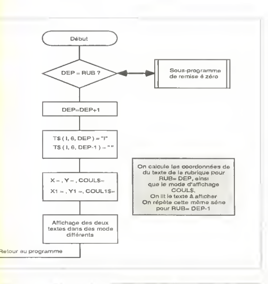
C'est un sous-programme qui sera chargé de gérer le déplacement vers le bas. Ce sous-programme consistera uniquement à lire les valeurs des éléments de la troisième dimension du tableau et à changer le mode d'affichage de deux rubriques adjacentes.

Les lignes du sous-programme

Ligne 1390
Teste si le mode vidéo inverse est positionné sur la dernière rubrique. Si c'est le cas, branchement au sous-programme de remise à zéro
Ligne 1400
La touche [**Flèche vers le bas**] a été appuyée. Le drapeau de déplacement est incrémenté d'une unité. Jusqu'à présent, c'était la première rubrique qui était affichée en mode vidéo inverse
Lignes 1410 à 1420
Il y a déplacement des modes d'affichage.
Lignes 1430 à 1480
On convertit les différentes coordonnées et on lit le mode d'affichage ainsi que le texte
Lignes 1490 à 1500
On écrit le texte en mode vidéo inverse.
Lignes 1510 à 1520
On écrit la rubrique précédente en mode normal

L'activation d'une rubrique

Un problème délicat en vérité. En effet, comme nous l'avons vu, l'activation d'une rubrique peut entraîner l'ouverture d'une fenêtre ou le lancement d'une application. Pour définir ces actions, nous avons introduit la notion de niveau de départ et niveau d'arrivée. Ici, nous nous occuperons uniquement du niveau d'arrivée
L'activation d'une rubrique se réalise en sélectionnant à l'aide de la touche [**Flèche vers le bas**] une rubrique puis en appuyant sur la touche [**ENVOI**]. Au départ, lors de la saisie des paramètres physiques des fenêtres, les niveaux



de départ et d'arrivée ont été définis Pour bien comprendre l'activation d'une rubrique, reportons-nous au synoptique général de l'application tel qu'il a été décrit quelques pages avant.

Plan d'action de la fenêtre 1

La rubrique 1 de la fenêtre 1 induit l'ouverture de la fenêtre 2 La rubrique 2 de la fenêtre lance l'exécution d'un programme La rubrique 3 de la fenêtre 1 induit l'ouverture de la fenêtre 3, la rubrique 4 de la fenêtre 1 induit la consultation du catalogue (Exelmémoire ou disquette), la rubrique 5 de la fenêtre 1 permet de quitter l'application en cours

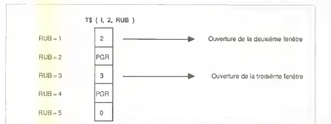
Traduisons en termes de niveaux d'arrivée.

Rubrique	Niveau
1	2
2	PGR
3	3
4	PGR / Consultation
5	0

Ces différents niveaux sont contenus dans les éléments 2 de la troisième dimension du tableau

Activation sur une fenêtre

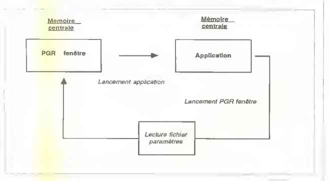
En fait, lorsque l'activation d'une fenêtre induit l'ouverture d'une fenêtre, le sous-programme chargé de l'activation lira les valeurs contenues

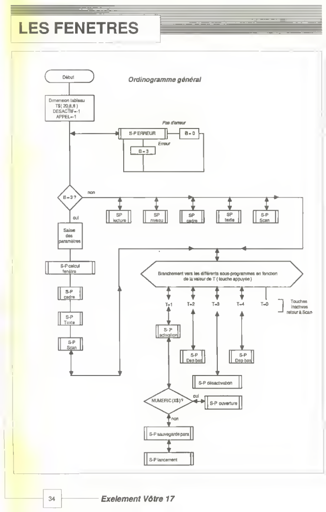


dans ces éléments. Ainsi, si la rubrique numéro 1 de la fenêtre numéro 1 est activée, le programme lira 2 comme valeur. Cette valeur sera prise comme le numéro d'ordre de la fenêtre à ouvrir. C'est simple comme bonjour. Un branchement simple mais judicieux à l'affichage d'une fenêtre et le tour est joué. Mais il existe quelques restrictions. En effet, il faut savoir discriminer les différents niveaux. Ainsi, si le niveau se résume au lancement du programme, cette opération devra être traitée à part. De même, si le niveau vaut zéro, ce qui correspond à l'abandon de l'application en cours, on devra également envisager un traitement différent.

Discrimination entre activation d'une fenêtre et lancement d'une application

Cette opération est facile à réaliser. Il suffit de tester si le niveau est une expression numérique, auquel cas, on a soit une ouverture de fenêtre soit un abandon. Si le niveau n'est pas une expression numérique, alors on procède au lancement d'un programme.
Comme nous traitons la différenciation entre l'activation qui entraîne l'ouverture d'une fenêtre et l'activation qui entraîne le lancement d'un programme, nous devons envisager tous les cas possibles.




Ordinogramme général
Début
Dimension tableau
T$(20,8,8)
DESACTIF=-1
APPEL=-1
S-P ERREUR
Pas d'erreur
B = 0
Erreur
B = 3
B = 3 ?
non
oui
SP lecture
SP niveau
SP cadre
SP texte
S-P Scan
Saisie des paramètres
S-P calcul fenêtre
S-P cadre
S-P Texte
S-P Scan
Branchement vers les différents sous-programmes en fonction de la valeur de T (touche appuyée)
T=1
T=2
T=3
T=4
T=0
Touches inactives retour à Scan
S- P activation
S-P Dep bas
S-P Dep bas
S-P désactivation
NUMERIC (X$)?
oui
S-P ouverture
non
S-P sauvegarde para
S-P lancement

Ordinogramme général (voir page précédente)

Le programme est développé sous forme de sous-programmes qui sont appelés au cours de son exécution
Plaçons-nous dans le cadre de l'utilisation du programme Au départ, les paramètres sont introduits A la fin de la saisie toutes les fenêtres crées disposent d'un jeu de paramètres qui sont:

I= Nombre de fenêtres, I varie entre 1 et NMAX

T$ (I, 0, 0)	Rubrique sélectionnée
T$ (I, 1, 0)	Position en X de la fenêtre
T$ (I, 2, 0)	Position en Y de la fenêtre
T$ (I, 3, 0)	Longueur de la fenêtre
T$ (I, 4, 0)	Hauteur de la fenêtre
T$ (I, 5, 0)	Nombre de rubriques
T$ (I, 6, 0)	Couleur de la fenêtre= C$

RUB= Nombre de rubriques

T$ (I, 1, RUB)	Niveau de départ
T$ (I, 2, RUB)	Niveau d'arrivée= X$
T$ (I, 3, RUB)	Position en X du texte de la rubrique
T$ (I, 4, RUB)	Position en Y du texte de la rubrique
T$ (I, 5, RUB)	Texte de la rubrique
T$ (I, 6, RUB)	Mode d'affichage= COUL$

Eléments particuliers

T$ (I,8,0)	Niveau de retour si la fenêtre est désactivée
T$ (I, 8, 1)	Fenêtres activées

A partir de ces éléments, toutes les possibilités peuvent être traitées. Prenons un exemple. Si l'activation de la rubrique 1 de la fenêtre 1 entraîne l'ouverture de la fenêtre 3 et que, l'activtion de la rubrique 2 de la fenêtre 3 entraîne l'ouverture de la fenêtre 5 Si on désactive la fenêtre 5, on doit revenir à la fenêtre 3 , de même la désactivation, de la fenêtre 3 implique le retour à la fenêtre 1 La désactivation des fenêtres dans ce cas ne peut être traitée de manière séquentielle

Lorsque la touche [ENVOI] d'une rubrique est activée, le programme appelle le sous-programme d'activation qui, au préalable passe dans un filtre afin de déterminer si le niveau d'arrivée est une expression numérique (ce qui entraîne l'ouverture d'une fenêtre) ou une chaîne de caractères (dans ce cas, il y a lancement d'une application).

Si l'activation d'une rubrique aboutit au lancement d'une application, tous les paramètres sont sauvegardés. En effet, l'application étant située sur un périphérique externe, toutes les données sont perdues

La sauvegarde des paramètres

En addition aux paramètres physiques à proprement parler, nous devons mettre à jour puis sauvegarder la valeur de désactivation pour la dernière fenêtre et les différentes fenêtres qui ont été activées avant le lancement de l'application De cette manière lorsque l'application est terminée, le programme de fenêtre est relancé et exploite le tableau principal qui doit être une représentation fidèle des différents états avant le lancement de l'application

```
100 '##################################
110 '#  PARAMETRAGE FENETRE           #
120 '##################################
130 CLS "GBB":CALL COLOR("GBB")
140 LOCATE (1,1):PRINT RPT$(" ",40)
150 DIM T$(20,8,8)
160 DESACTIF=-1:APPEL=1
170 ON ERROR 2540
180 '##################################
190 '#  APPEL S-P DE LECTURE          #
200 '##################################
210 CLS "WBB":CALL COLOR("WBL")
220 CALL SPEECH "S,00B6FF")
230 LOCATE (11,10):PRINT "EExxEECCUUTTIIOONN"
240 LOCATE (12,13):PRINT "EENN  CCOOUURRSS"
250 GOSUB 2160:GOSUB 1090
260 CLS "GBE":CALL COLOR("GB")
270 LOCATE (5,5):PRINT "Nombre de fenetres:";
280 ACCEPT BEEP VALIDATE("12345678")SIZE(1)NULL(1),NMAX
290 CALL LOAD("INIT.COM")
300 CALL DO("INIT.COM",NMAX,ADD)
310 '##################################
320 '#      SAISIE DES PARAMETRES     #
330 '##################################
340 FOR I=1 TO NMAX
350 CLS
360 LOCATE (5,5):PRINT "Position en x:";
370 ACCEPT BEEP VALIDATE(DIGIT)SIZE(2)NULL(1),T$(I,1,0)
380 X=VAL(T$(I,1,0))
390 IF X<=0 THEN 360
400 IF X>39 THEN 360
410 LOCATE (6,5):PRINT "Position en y:";
420 ACCEPT BEEP VALIDATE(DIGIT)SIZE(2)NULL(1),T$(I,2,0)
430 Y=VAL(T$(I,2,0))
440 IF Y<=0 THEN 410
450 IF Y>20 THEN 410
460 LOCATE (7,5):PRINT "Longueur:";
470 ACCEPT BEEP VALIDATE(DIGIT)SIZE(2)NULL(5),T$(I,3,0)
480 L=VAL(T$(I,3,0))
490 IF L>40-X THEN 460
500 LOCATE (8,5):PRINT "Couleur:";
```

LES FENETRES

Mise à jour des fenêtres activées et des valeurs de désactivation

Le numéro de la fenêtre que l'on quitte est égal à T$ (FEN, 1, DEP) où FEN = I Cette information nous sera utile lorsque nous lancerons une application Ce numéro de fenêtre est affecté à la variable ANCIEN

1 Dans le cas d'une ouverture de fenêtre

Pour tout ce qui concerne l'avtivation, l'ouverture la désactivation d'une fenêtre, le numéro de la fenêtre en cours de traitement est égal à FEN.

Le numéro de la fenêtre activée est rangé dans l'élément T $ (I, 8, 1) où I=FEN. Cette information sera utilisée lors de la restitution des différents états après le lancement d'une application. La position du curseur (la rubrique activée) de la fenêtre précédente est stocké dans T$ (FEN, 2, DEP). Le niveau d'appel de la fenêtre qui sera activée est rangé dans l'élément T$ (FEN, 8, 0) Si la valeur de la nouvelle fenêtre est égale à 0, le programme est fini.

2- Dans le cas d'un lancement de programme

Le numéro de la dernière fenêtre activée ainsi que la dernière rubrique sélectionnée sont rangés dans les éléments T$ (ANCIEN, 8, 1) et T$ (ANCIEN, 0, 0) Le niveau d'appel est sauvegardé dans un fichier Lorsque ce fichier est écrit (sur Exelmémoire), le numéro de la dernière fenêtre activé prend la valeur de APPEL et le sous-programme chargé de la sauvegarde de tous les paramètres est appelé Une fois cette opération réalisée l'application est lancée.

Les sous-programmes intéressés se situent respectivement aux lignes 1680, 1750, 1850 et 2020

Lancement d'une application

Le lancement d'une application se réalise simplement **RUN "2."&X$.** N'oublions pas que X$ est chargé avec la valeur du niveau d'arrivée X$ dans ce cas est une expression chaîne de caractères représentant un nom de programme Attention, le nom du programme doit être inscrit dans le catalogue de l'Exelmémoire ou de l'Exeldisk

```
[illegible] BEEP VALIDATE "BbW,GRMC"(SIZE [illegible]
[illegible] LOCATE (9,5):PRINT "Nombre de rubriques: ";
[illegible] BEEP VALIDATE("12345" SIZE 1)NULL [illegible]
[illegible] RUB=VAL(T$(I,5, [illegible]
[illegible] HH=VAL(T$(I,5, [illegible])+2
[illegible] T$(I,4,0)=STR$(HH)
[illegible]
590 FOR [illegible] TO RUB
600 LOCATE (5,5):PRINT "Te [illegible]
610 ACCEPT BEEP VALIDATE ALFHA)SIZE(L)NULL [illegible]
620 T$(I,1,2)=STR$ I
630 LOCATE (6,5):PRINT "Niveau départ [illegible]
640 PRINT I
650 LOCATE (7,5):PRINT "Niveau d'arrivée:";
655 ACCEPT BEEP SIZE [illegible] NULL [illegible]
[illegible] IF NOT [illegible] T$ [illegible],2,1))THEN 720
[illegible]
[illegible] THEN 730
[illegible] NIVEAU)=-1 THEN 650
[illegible] EAU NMAX THEN 650
[illegible] IF NIVEAU =I THEN [illegible]
[illegible] T$(I,7,J)=STR$ [illegible]
[illegible] T$(I,4,1)=STR$ [illegible]
[illegible]
[illegible] NEXT I
[illegible] '################################
[illegible] '#   REMPLISSAGE DE [illegible]    #
[illegible] '################################
[illegible] CLS
[illegible] FOR [illegible]
[illegible] LOCATE (I,[illegible]):PRINT RPT$([illegible],40)
[illegible] NEXT
[illegible] '#######GESTION DES FENETRES#####
[illegible] GOSUB 900
[illegible] '################################
[illegible] '#   AFFICHAGE DE LA FENETRE 1   #
[illegible] '################################
[illegible] FEN=1
[illegible] RUB=VAL(T$(FEN,5,0))
950 FOR Z=1 TO RUB:T$(FEN,6,Z)="":NEXT:T$(FEN,[illegible]
960 X=VAL(T$(FEN,1,0))
970 Y=VAL(T$(FEN,2,0))
980 L=VAL(T$(FEN,3,0))
990 HH=VAL(T$(FEN,4,0))
1000 C$=T$(FEN,6,0)
1010 GOSUB 1170
1020 GOSUB 1240
1030 GOSUB 1050
1040 RETURN
1050 '################################
1060 '#     ATTENTE TOUCHE            #
1070 '################################
1080 DEP=1
1090 CALL KEY1(A,B)
1100 IF A=255 THEN 1090
1110 CALL SPEECH("S,00B2FF")
1120 T=-1*(A=13)-2*(A=130)-3*(A=27)-4*(A=128)
1130 IF T=0 THEN 1090
1140 ON T GOSUB 1680,1760,2720,1760
1150 GOTO 1090
1160 RETURN
```

LES FENETRES

```
1[illegible] '################################
1[illegible] '#   ROUTINE [illegible]         #
1[illegible] '################################
1[illegible] CALL LOAD "DFEN.COM")
1[illegible] [illegible]
1[illegible] CALL DO("OPEN.COM",ADD,X,Y,L,HH,C$)
1[illegible] RETURN
1240 '################################
1250 '#    ECRITURE DES TEXTES        #
1[illegible] '################################
1[illegible] RUB=VAL(T$(FEN,5,[illegible]))+1
1[illegible] FOR J=1 TO RUB
1[illegible] Y=Y+1
1[illegible] LOCATE [illegible]:COUL$=T$(FEN,6,J):C$=SEG$(C$,1,1)&COUL$
171[illegible] Y$=SEG$(C$,2,1):IF Y$=" " THEN C$=SEG$(C$,1,1)
172[illegible] CALL COLOR([illegible]C$)
17[illegible] LOCATE (Y,X):PRINT T$(FEN,5,J)
174 NEXT J
1[illegible] RETURN
1[illegible] '################################
  '#  [illegible] DEPLACEMENT VERS BAS  #
1[illegible] '################################
1[illegible] IF DEF=RUB THEN GOSUB 1540:RETURN
14[illegible] DEF=DEF+1
141 T$(FEN,6,DEF)="1"
14[illegible] T$(FEN,6,DEF-1)="
14[illegible] X=VAL(T$(FEN,3,DEF)):Y=VAL(T$(FEN,4,DEF))
144[illegible] COUL$=T$(FEN,6,DEF)
145[illegible] X1=VAL(T$(FEN,3,DEF-1)):Y1=VAL(T$(FEN,4,DEF-1))
146 COUL1$=T$(FEN,5,DEF-1)
14[illegible] TEXTE$=T$(FEN,5,DEF)
14[illegible] TEXTE1$=T$(FEN,5,DEF-1)
14[illegible] CALL COLOR([illegible]C$&COUL$)
1[illegible] LOCATE (Y,X):PRINT TEXTE$
1[illegible] CALL COLOR([illegible])
1[illegible] LOCATE (Y1,X1):PRINT TEXTE1$
1[illegible] RETURN
1540 '################################
1550 '#  REMISE A BLANC               #
1560 '################################
1[illegible] DEF 1:T$(FEN,6,DEF)="1"
1[illegible] [illegible]
1[illegible] [illegible] = [illegible](FEN,4,DEF):COUL$=T$(FEN,6,DEF)
1[illegible] TEXTE$=T$(FEN,5,DEF)
1[illegible] CALL COLOR([illegible])
1[illegible] LOCATE [illegible]:TEXTE$
1[illegible] [illegible] 1=[illegible]T$(FEN,4,RUB)
1[illegible] TEXTE1$=T$(FEN,[illegible],RUB)    :COUL1$=T$(FEN,6,RUB)
1[illegible] CALL COLOR([illegible]C$&COUL1$)
1[illegible] LOCATE [illegible]:PRINT TEXTE1$
1[illegible] RETURN
1[illegible] '################################
1[illegible] '# SOUS-PROGRAMME [illegible]ATION   #
  '################################
1[illegible] [illegible](FEN,2,DEF):ANCIEN=VAL(T$(FEN,1,DEF))
1[illegible] [illegible] THEN GOSUB 1750 ELSE GOSUB 1850
1[illegible] [illegible]
1[illegible] RETURN
1750 '################################
1760 '#    OUVERTURE FENETRE          #
1770 '################################
1780 I=FEN:T$(I,8,1)=STR$(FEN)
1790 T$(FEN,[illegible])=STR$[illegible] 'ANCIEN CURSEUR
1800 FEN=VAL(T$(FEN,2,DEF)) 'NOUVELLE FENETRE
18[illegible] T$(FEN,8,[illegible])=STR$(I)'SAUVEGARDE NIVEAU D APPEL
1820 DEF=1
1830 IF FEN=0 THEN END
1840 RETURN
```

Retour aux fenêtres

Lorsqu'une application est lancée, le programme de gestion de fenêtres est remplacé dans la mémoire centrale de l'ordinateur. Le problème est évident : comment revenir au niveau d'appel lorsque l'application a accompli sa tâche

En fait, la dernière ligne du programme d'application doit contenir une instruction de lancement du programme des fenêtres. Lorsque le programme général des fenêtres s'exécute, une instruction de lecture des deux fichiers (APPEL et PARA) et exécutée. Les différents paramètres sont lus puis le programme appelle des sous-programmes chargés de rétablir l'état de l'écran avant le lancement de l'application

Lorsque l'application lancée par l'activation d'une rubnque d'une fenêtre termine sa tâche, une ligne d'instruction **RUN "2.FENETRE"** (si le programme des fenêtres s'appelle FENETRE) chasse de la mémoire centrale l'application et recharge puis lance le programme FENETRE. A ce moment précis doit intervenir un branchement entre la création de nouvelles fenêtres et l'exploitation des fenêtres définies précédemment

Tout le problème réside dans la présence ou non d'un fichier de sauvegarde des paramètres. En effet, si le fichier PARA existe, ce dernier doit être lu puis le programme doit réafficher la séquence des fenêtres jusqu'à l'appel du lancement de l'application. Par contre si le fichier PARA n'existe pas, on peut conclure que nous sommes dans une phase de création de fenêtres. Comment traiter ce problème ? En fait, nous allons procéder en nous aidant d'une éventuelle erreur de lecture.

Nous savons que lorsque un programme demande la lecture d'un fichier qui n'existe pas, une erreur est générée. Or, en reconnaissant l'erreur, nous pouvons affirmer à coup sûr que le programme doit commencer une phase de création. Dans cette hypothèse, nous écrirons au tout début du programme une ligne d'instruction faisant appel à une procédure d'erreur **ON ERROR *ligne*.** Ensuite on appelle un sous-programme de lecture. Si une erreur de lecture intervient (fichier PARA inexistant), la routine de gestion des erreurs retournera le programme à la phase de création. Si le fichier PARA existe, ce dernier est lu puis le programme retourne à la phase d'affichage des fenêtres.

LES FENETRES

Affichage des fenêtres avant le lancement de l'application

La première phase consiste à lire les deux fichiers PARA et APPEL. Cette opération se réalise dans le sous-programme de lecture situé à partir des lignes 2160. Après cette phase de lecture des paramètres, le sous-programme d'ouverture sur le niveau d'appel est demandé. Le sous-programme est situé à partir de la ligne 2340.

Le sous-programme d'ouverture sur le niveau d'appel

Ce sous-programme permet de restituer l'écran tel qu'il était avant le lancement de l'application. Dans une boucle variant de 1 jusqu'à APPEL (dernière fenêtre activée avant le lancement de l'application) avec comme variable de boucle FEN (FEN représente toujours une fenêtre en cours de traitement), tous les paramètres sont convertis. Ces paramètres au risque de nous répéter sont X, Y, L, HH, Couleur, RUB= nombre de rubrique, DEP=dernière rubrique activée. Une ligne d'instructions rejette de l'affichage les fenêtres qui n'ont pas été activées.

ACTIVE$=T$(FEN,8,1): IF ACTIVE$="@" THEN NEXT

Cette ligne (2420) élimine de la séquence d'affichage les fenêtres qui n'ont pas été activées. En effet dans la saisie des paramètres acune fenêtre n'est activée au départ. Pour matérialiser ce fait, nous avons forcé les élémants du tableau correspondant avec la chaîne de caractères @

Une fois les paramètres convertis, les textes des différentes fenêtres sont affichés ainsi que leur cadre. L'affichage des textes est réalisé par le sous-programme situé à la ligne 2610. Lorsque tous les éléments des fenêtres sont affichés, le sous-programme de saisie des touches de commande est appelé et de nouvelles manipulations peuvent être effectuées.

Vous noterez au passage que l'ouverture des fenêtres est toujours accompagnée par l'appel aux routines assembleur.

La désactivation d'une fenêtre

La désactivation des fenêtres est une opération délicate. En effet, elle ne peut être traitée de manière séquentielle. Comme nous l'avons vu dans les précédentes sections, la succession des

```
1850 '##################################
1860 '# LANCEMENT PROGRAMME             #
1870 '##################################
1880 T$(ANCIEN,8,1)=STR$(ANCIEN)
1890 'BREAK
1900 APPEL=ANCIEN:T$(APPEL,0,0)=STR$(DEF)
1910 OPEN #1,"2.APPEL",OUTPUT
1920 PRINT #1,APPEL
1930 CLOSE #1
1940 CLS "Abb":CALL COLOR("1WbL"):LOCATE (8,11):PRINT
     "EEXXEECCUUTTIIOONN
1950 LOCATE (11,11):PRINT  EENN  CCOOUURRSS"
1960 CALL SPEECH("S,00B6FF")
1970 GOSUB 2030
1980 FEN=APPEL
1990 ON ERROR 2540
2000 RUN "2."+X$
2010 RETURN
2020 '##################################
2030 '#    S-P SAUVEGARDE              #
2040 '##################################
2050 OPEN #1,"2.PARA",OUTPUT
2060 PRINT #1,NMAX
2070 FOR I=1 TO NMAX
2080 FOR J=0 TO 8
2090 FOR K=0 TO 8
2100 IF LEN(T$(I,J,K))=0 THEN PRINT #1,"@" ELSE PRINT #1,T$(I,J,K)
2110 NEXT K
2120 NEXT J
2130 NEXT I
2140 CLOSE #1
2150 RETURN
2160 '##################################
2170 '#   SOUS-PROGRAMME LECTURE       #
2180 '##################################
2190 OPEN #1,"2.PARA",INPUT
2200 INPUT #1,NMAX
2210 FOR I=1 TO NMAX
2220 FOR J=0 TO 8
2230 FOR K=0 TO 8
2240 INPUT #1,T$(I,J,K)
2250 NEXT K
2260 NEXT J
2270 NEXT I
2280 CLOSE #1
2290 OPEN #1,"2.APPEL",INPUT
2300 INPUT #1,APPEL
2310 CLOSE #1
2320 GOSUB 2340
2330 RETURN
2340 '##################################
2350 '#   OUVERTURE SUR NIVEAU         #
2360 '##################################
2370 CLS "6BB":CALL COLOR("0GB")
2380 CALL LOAD("INIT.COM")
2390 CALL DO("INIT.COM",NMAX,ADD)
2400 IF APPEL=1 THEN GOSUB 930
2410 FOR FEN=1 TO APPEL
2420 ACTIVE$=T$(FEN,8,1):IF ACTIVE$="@" THEN 2510
2430 X=VAL(T$(FEN,1,0))
2440 Y=VAL(T$(FEN,2,0))
2450 L=VAL(T$(FEN,3,0))
2460 HH=VAL(T$(FEN,4,0))
2470 DEP=VAL(T$(FEN,0,0))
2480 C$=T$(FEN,6,0):RUB=VAL(T$(FEN,5,0))
2490 GOSUB 1170
2500 GOSUB 2600
2510 NEXT FEN
2520 FEN=APPEL
2530 RETURN
```

```
[illegible] '################################
[illegible] '#   SOUS-PROGRAMME ERREUR        #
[illegible] '################################
    CALL [illegible]
    [illegible] (2840)
    [illegible]
    ################################
[illegible] #   AFFICHE LES TEXTES           #
    ################################
[illegible] FOR [illegible]
[illegible] [illegible]:[illegible]=VAL(T$(FEN,4,J))
[illegible] [illegible]=T$(FEN,[illegible],J):IF COUL$="@" THEN COUL$=""
[illegible] [illegible]=T$(FEN,5,J):COUL$=T$(FEN,6,J):IF
    COUL$="@" THEN COUL$=""
[illegible] LOCATE [illegible]:C$=SEG$(C$,1,1)&COUL$:CALL COLOR("0"&C$)
[illegible] [illegible] TEXTE$
[illegible] NEXT [illegible]
    [illegible]=SEG$(C$,1,[illegible])
    RETURN
    '################################
[illegible] '#   FERMETURE FENETRE            #
[illegible] '################################
[illegible] [illegible]=1 THEN DEF=VAL(T$(FEN,[illegible])):RETURN
[illegible] T$(FEN,[illegible])=[illegible]
    DESACTIF=[illegible] T$(FEN,[illegible])
[illegible] [illegible] T$(DESACTIF,[illegible]):RUB=VAL(T$(DESACTIF,5,0))
[illegible] [illegible]=T$(DESACTIF,[illegible]):CALL COLOR("0"&C$)
[illegible] [illegible] LOADSE.COM [illegible]
[illegible] CALL [illegible].COM ,ADD)
[illegible] [illegible]=DESACTIF
[illegible] RETURN
[illegible] ################################
[illegible] '#   AFFICHAGE DES ERREURS        #
[illegible] ################################
    CALL [illegible]
    CALL [illegible]("S,00B6FF")
[illegible] LOCATE 19,[illegible]:PRINT "[Fichier non trouvé]"
[illegible] PAUSE
[illegible] RETURN
```

différentes fenêtres ne se réalise pas de manière linéaire. La fenêtre 1 peut appeler la fenêtre 3 qui peut appeler la fenêtre 5 qui lance une application. Le sous-programme chargé de la fermeture des fenêtres est placé à partir de la ligne 2720. Lorsqu'une fenêtre est désactivée, l'élément représentant les fenêtres activées est forcé avec la chaîne de caractères @. La valeur de la dernière fenêtre activée est chargé dans la variable DESACTIF. La dernière rubrique activée, le nombre de rubrique, la couleur de la dernière fenêtre sont chargés dans DEP, RUB et C$. Les routines assembleurs sont appelées et le programme de saisie des touches de commande est appelé. Tous les paramètres nécessaires au bon fonctionnement du déplacement, de l'activation ou d'une autre désactivation sont transmis.

Une application type

Tapez le listing final, sauvegardez-le dans l'Exelmémoire sous le nom LST3. Videz l'Exelmémoire de tout fichier ou programme parasite (mis à part les programmes assembleurs et les applications nécessaires au bon fonctionnement des fenêtres).

Ecriture des applications

Si vous désirez lancer des programmes à partir de fenêtres, écrivez-les. Les programmes doivent obligatoirement se terminer par une instruction RUN "2.LST3". Voir exemple listing 6.

Chargez le programme des fenêtres puis exécutez-le. Le programme commence la phase de saisie des paramètres

Nombre de fenêtres: 5 **[ENVOI]**

Position en X: 1 **[ENVOI]**
Position en Y: 1 **[ENVOI]**
Longueur: 5 **[ENVOI]**
Couleur: W **[ENVOI]**
Nombre de rubriques: 5 **[ENVOI]**

L'écran se vide puis affiche

Texte: **[ENVOI]**
(le texte dans ce cas précis se résume à la lettre A)
Niveau de départ: 1
(le niveau de départ est toujours le numéro de la fenêtre en cours)
Niveau d'arrivée: 2 **[ENVOI]**
(ouverture de la deuxième fenêtre)

Texte **[ENVOI]**
Niveau de départ:1
Niveau d'arrivée: EX1 **[ENVOI]**
(lancement application)

Texte **[ENVOI]**
Niveau de départ 1
Niveau d'arrivée: 3 **[ENVOI]**
(ouverture de la troisième fenêtre)

Texte: **[ENVOI]**
Niveau de départ: 1
Niveau d'arrivée : EX2 **[ENVOI]**
(lancement d'une application chargée de lire le catalogue CRAM)

LES FENETRES

Texte [ENVOI]
Niveau de départ: 1
Niveau d'arrivée 0 [ENVOI]
(abandon du programme en cours)

Fenêtre numéro 2

Position en X 2	[ENVOI]
Position en Y 2	[ENVOI]
Longueur. 5	[ENVOI]
Couleur. C	[ENVOI]
Nombre de rubriques 2	[ENVOI]

L'écran se vide puis affiche

Texte [ENVOI]
(le texte dans ce cas précis se résume à la lettre A)
Niveau de départ 2
(le niveau de départ est toujours le numéro de la fenêtre en cours)
Niveau d'arrivée 4 [ENVOI]
(ouverture de la quatrième fenêtre)

Texte [ENVOI]
Niveau de départ. 2
Niveau d'arrivée 0 [ENVOI]
(abandon du programme en cours)

Fenêtre numéro 3

Position en X: 3	[ENVOI]
Position en Y 3	[ENVOI]
Longueur 5	[ENVOI]
Couleur R	[ENVOI]
Nombre de rubriques 5	[ENVOI]

L'écran se vide puis affiche

Texte [ENVOI]
(le texte dans ce cas précis se résume à la lettre A)
Niveau de départ 3
(le niveau de départ est toujours le numéro de la fenêtre en cours)
Niveau d'arrivée EX2 [ENVOI]
(lancement d'une application)

Texte [ENVOI]
Niveau de départ 3
Niveau d'arrivée EX3 [ENVOI]
(lancement application)

Texte [ENVOI]
Niveau de départ:3
Niveau d'arrivée EX4 [ENVOI]
(Lancement d'une application)

Texte [ENVOI]
Niveau de départ 3
Niveau d'arrivée : EX5 [ENVOI]
(lancement d'une application)

Texte [ENVOI]
Niveau de départ: 3
Niveau d'arrivée EX6 [ENVOI]
(Lancement d'une application)

Fenêtre numéro 4

Position en X 4	[ENVOI]
Position en Y 4	[ENVOI]
Longueur 5	[ENVOI]
Couleur Y	[ENVOI]
Nombre de rubriques 3	[ENVOI]

L'écran se vide puis affiche

Texte: [ENVOI]
(le texte dans ce cas précis se résume à la lettre A)
Niveau de départ 4
(le niveau de départ est toujours le numéro de la fenêtre en cours)
Niveau d'arrivée 5 [ENVOI]
(ouverture de la cinquième fenêtre)

Texte [ENVOI]
Niveau de départ 4
Niveau d'arrivée EX7 [ENVOI]
(lancement application)

Texte [ENVOI]
Niveau de départ 4
Niveau d'arrivée EX8 [ENVOI]
(ouverture de la troisième fenêtre)

Fenêtre numéro 5

Position en X: 5	[ENVOI]
Position en Y 5	[ENVOI]
Longueur. 5	[ENVOI]
Couleur M	[ENVOI]
Nombre de rubriques 2	[ENVOI]

L'écran se vide puis affiche

Texte [ENVOI]
(le texte dans ce cas précis se résume à la lettre A)
Niveau de départ 5
(le niveau de départ est toujours le numéro de la fenêtre en cours)
Niveau d'arrivée: EX8 [ENVOI]
(ouverture de la deuxième fenêtre)

L'écran se vide et la première fenêtre s'affiche Manoeuvrez les différentes touches de commandes et vérifiez le bon fonctionnement

Les applications doivent obligatoirement figurer dans le catalogue de l'Exelmémoire et ces dernières doivent impérativement se terminer par le lancement du programme pnncipal des fenêtres (LST3).

Codage dee programmes essembleur soue BASIC

Si vous ne possédez pas le langage EXELMAX, vous devez à partir du BASIC créer les programmes INIT.COM, OPEN.COM, CLOSE.COM. Cette opération est fastidieuse, aussi, faites bien attention à ce que vous tapez La moindre erreur est fatale et ne pardonne pas

```
      CREA: création de fich.ers .COM
   '  vers 0 SH 21127
   '
140 CLS "6bB":CALL COLOR("0YB"):DIM F$(2),V(2)
15  RESTORE 9000:FOR I=0 TO 2:READ F$(I),V(I):NEXT' label, ligne aff
    RESTORE [illegible]:FOR FIC=  TO 2
    READ A$,NBR:OPEN #2,F$(FIC),OUTPUT,VARIABLE NBR
    LOCATE  V(FIC),.):PRINT F$(FIC)+RPT$(" ",23-LEN(F$(FIC)));"  "&STR$(NBR)
    FOR I=0 TO NBR-1:READ A0:LOCATE (V(FIC)),20):PRINT STR$(1+I)
    PRINT #2,CHR$(A0);:NEXT:PRINT #2,  :CLOSE #2:CALL SPEECH("S,00E2FF")
    NEXT:END
    DATA 2.INIT.COM,16
    DATA 2.OPEN.COM,16
    DATA 2.CLOSE.COM,[illegible]
      INIT
    DATA DEBUT,144
    DATA 4,128,10,15,0,77,[illegible],1,72,2,197,[illegible]8,192,2,197
    DATA [illegible]6,28,217,[illegible]5,0,121,16,28,[illegible]2,28,222,28,21[illegible],28,216
    DATA [illegible]8,[illegible],122,[illegible]92,124,74,128,140,15,6,114,[illegible],128,142
    DATA 18,46,114,176,82,74,128,142,19,44,114,18,156,185,201
    DATA 1[illegible],208,0,2,170,18,208,1,2,170,16,50,200,184,74
    DATA 128,14[illegible],13,[illegible],1,4,74,128,142,18,12,114,79,20,152,74
    DATA 128,142,19,42,114,74,128,142,18,44,114,46,130,28,18
    DATA 46,1[illegible],181,247,14,70,152,[illegible],69,152,74,128,142,18,56
    DATA 114,71,79,152,78,0,121,74,2,120,76,227,79,222,176
    DATA 28,79,152,92,128,142,7,128,14[illegible]
      OPEN
    DATA DEBUT,52
    DATA 4,128,72,96,160,192,64,128,224,0,1,7,5,6,2
    DATA 4,[illegible],0,82,89,77,67,71,98,87,66,10,46,146,46
    DATA 1[illegible],16,50,249,[illegible],218,45,146,46,1[illegible],14,50,1[illegible],2,122
    DATA 1[illegible],25,66,46,14[illegible],46,1[illegible],22,18,15,50,14,0,114,16
    DATA 7,[illegible],136,7,224,14,4,114,16,[illegible],8,136,16,224,14
    DATA 4,114,16,[illegible],9,126,196,129,140,240,25,218,247,14,20
    DATA 152,19,0,12[illegible],20,82,12[illegible],205,129,142,247,14,20,152,19
11090 DATA [illegible],12[illegible],20,82,12[illegible],25,2,122,187,129,142,247,14,20,152
    DATA 22,0,68,1,130,170,204,224,249,202,4,226,241,129,173
111   DATA [illegible],82,185,22,0,68,249,129,170,222,224,185,249,202,5
1120  DATA 226,241,129,17[illegible],8,82,22,24,114,8,36,128,40,128,6
11130 DATA 14,184,8,36,128,40,128,6,14,232,22,1,114,4,226
11140 DATA 2,45,8,36,128,40,128,6,14,246,14,81,152,247,128
11150 DATA 142,18,48,114,247,128,142,18,44,114,10,15,0,73,16
```




```
11160 DATA 1,72,2,193,136,192,3,197,128,28,217,15,0,121,16
11170 DATA 28,72,28,222,28,210,26,216,16,1,152,27,82,124,247
11180 DATA 128,140,18,6,114,247,128,142,18,46,114,176,82,247,128
11190 DATA 142,18,44,114,15,156,185,201,17,208,0,2,170,18,208
11200 DATA 1,2,170,18,50,200,184,19,210,25,218,245,14,22,152
11210 DATA 21,0,121,22,0,72,190,26,18,15,0,121,16,82,120
11220 DATA 235,18,218,46,130,8,36,128,40,128,6,14,46,130,8
11230 DATA 36,128,40,126,6,14,18,26,66,15,0,121,16,211,21
11240 DATA 0,121,22,4,120,46,130,25,18,46,130,26,18,46,130
11250 DATA 27,18,46,130,28,18,8,36,128,40,128,6,14,245,14
11260 DATA 22,152,16,219,20,16,152,29,129,142,10,46,130,21,18
11270 DATA 46,130,22,18,247,14,1,152,29,25,70,56,0,40,2
11280 DATA 88,191,207,176,181,192,178,46,130,179,8,36,128,40,128
11290 DATA 6,14,246,14,30,152,10,247,128,140,18,60,114,67,129
11300 DATA 142,9,226,38,175,125,25,217,154,129,142,25,216,117,128
11310 DATA 142,22,69,152,247,128,142,18,56,114,71,1,152,0,41
11320 DATA 4,88,25,28,190,26,18,272,22,127,114,181,5,271,24
11330 DATA 0,77,8,36,128,40,128,6,14,24,208,8,36,128,40
11340 DATA 128,6,14,246,14,30,152,25,79,66,9,129,142,26,79
11350 DATA 66,9,129,142,27,79,66,9,129,142,28,79,66,9,129
11360 DATA 142,30,79,152,9,129,142,7,128,140
12000 ' CLOSE
12010 DATA DEBUT,265
12020 DATA 4,128,10,15,0,72,16,1,72,2,193,138,192,3,193
12030 DATA 178,28,217,15,0,121,16,28,72,28,222,28,210,28,216
12040 DATA 16,1,152,27,82,124,195,128,140,18,6,114,195,128,142
12050 DATA 18,46,114,176,82,195,128,142,18,44,114,18,156,185,201
12060 DATA 17,208,0,2,170,18,208,1,2,170,18,50,200,184,195
12070 DATA 128,140,18,54,114,69,30,152,210,25,218,15,0,121,16

12080 DATA 0,72,190,26,18,14,22,152,21,0,121,22,82,120,275
12090 DATA 18,218,46,120,8,36,128,40,128,6,14,46,130,8,36
12100 DATA 128,40,128,6,14,18,26,66,245,22,14,152,16,22,152
12110 DATA 21,0,121,22,4,120,14,16,152,233,128,142,25,208,8
12120 DATA 36,128,40,128,6,14,26,208,8,36,128,40,128,6,14
12130 DATA 27,208,8,36,128,40,128,6,14,28,208,8,36,128,40
12140 DATA 128,6,14,246,30,22,152,14,22,152,21,208,8,36,128
12150 DATA 40,128,6,14,22,208,8,36,128,40,128,6,14,239,18
12160 DATA 218,8,36,128,40,128,6,14,8,36,128,40,128,6,14
12170 DATA 18,208,19,226,185,8,36,128,40,128,6,14,184,46,130
12180 DATA 178,10,1,230,176,8,36,128,40,128,6,14,246,14,30
12190 DATA 152,30,79,152,217,128,142,7,128,140
```

TAROTS

Une magnifique simulation du jeu de terots

```
1 DIM F(170)
4 L=0:CLS "bBB"
5 LOCATE (10,4):PRINT "MICHEL MOUPS!   CLERMONT FD  10 12 1985":PAUSE 4
6 REM DESSIN
8 REM CARACTERES ATOUTS
10 CALL CHAR(1,"FF457FFAEDCF878909978")
12 CALL CHAR(2,"FF8576C5EFFAAACBE5C1")
14 CALL CHAR(3,"FF422231187AAAB8CFE1")
16 CALL CHAR(4,"123FEFELACADD76C42FF")
18 CALL CHAR(5,"0780905047E25F772AFF")
20 CALL CHAR(6,"0204077050604000A0FF")
22 REM COEUR
24 CALL CHAR(10,"00006CFEFE7C38101000")
26 CALL CHAR(11,"000000000000006CFEFE")
28 CALL CHAR(12,"7C381010")
30 REM PIQUE
32 CALL CHAR(20,"00101038 7CFEFE381078")
34 CALL CHAR(21,"000000000000101038 7C")
36 CALL CHAR(22,"FEFE381038")
38 REM CARREAU
40 CALL CHAR(30,"081C3E7F3E1C08")
42 CALL CHAR(31,"000000000000081C3E7F")
44 CALL CHAR(32,"3E1C08")
46 REM TREFLE
48 CALL CHAR(40,"1038 1054FE5410103800")
50 CALL CHAR(41,"000000000010381054FE")
52 CALL CHAR(42,"54101038 0000000000000")
54 REM VALET
56 CALL CHAR(90,"0060F0F0780C1E1E7E7E")
58 CALL CHAR(91,"000000000000010101FF")
60 CALL CHAR(92,"7E3E007E7EE6868687FF")
62 CALL CHAR(93,"000000000000003F0FF")
64 CALL CHAR(94,"FF0F0C")
66 CALL CHAR(95,"FFE161616 77E7C07C7C")
68 CALL CHAR(96,"FF808080")
70 CALL CHAR(97,"7C7C7878 01E0F0F06")
72 REM DAME
74 CALL CHAR(110,"00003C7EFFFFDB997C7E")
76 CALL CHAR(111,"00010103020301000FF")
78 CALL CHAR(112,"FFFFFF7E7E7EFFBD3CFF")
80 CALL CHAR(113,"00808 0C040C000000FF")
82 CALL CHAR(114,"FF00000102020 30101")
84 CALL CHAR(115,"FF3CBDFF7E7E7EFFFFFF")
86 CALL CHAR(116,"FF00008 0C040C08080")
88 CALL CHAR(117,"FF3CDBDBFFFF7E18")
90 REM CAVALIEF
92 CALL CHAR(100,"00000000070702072 7B")
94 CALL CHAR(101,"0000070F0F141404 04FF")
96 CALL CHAR(102,"7777FEFCFCFE050504FF")
98 CALL CHAR(103,"C0C00000000000000FF")
100 CALL CHAR(104,"FF00000000000000303")
102 CALL CHAR(105,"FF20A0A07F3F3F7FEEEE")
104 CALL CHAR(106,"FF20202828F0F0E0")
106 CALL CHAR(107,"DE4E040E0E")
108 REM ROI
110 CALL CHAR(120,"000000000000547C7C54")
112 CALL CHAR(121,"0000081C1D0908 0E0CFF")
114 CALL CHAR(122,"7C3810FFFFFEFE7C7CFF")
116 CALL CHAR(123,"1C083E080 8FC7C0808FF")
118 CALL CHAR(124,"FF10103E3F11107C1038")
120 CALL CHAR(125,"FF3E3E7F7FFFFF081C3E")
122 CALL CHAR(126,"FF3070D090B83810")
124 CALL CHAR(127,"2A3E3E2A")
126 REM EXCUSE
128 CALL CHAR(50,"0000000000000C0E0303")
130 CALL CHAR(51,"00000070386060E0C0E0")
132 CALL CHAR(52,"00000206010304")
134 CALL CHAR(53,"030247BE0E8CDCD83830")
136 CALL CHAR(54,"3018")
138 CALL CHAR(55,"000000000000101 0303")
140 CALL CHAR(56,"70286C44C2C08080
142 CALL CHAR(57,"03060404")
144 REM TITRE
145 CALL HRON("B",8,6):GOSUB 382
148 DATA 40,80,100,120,140,176,204,226,250,290,400
150 DATA 40,60,100,120,140,180,200,240,250,290,400
```

```
150 DATA 40,50,70,100,140,160,170,180,200,210,220,240,250,290,400
154 DATA 50,70,90,100,120,124,140,150,170,180,200,210,220,240,260,280,400
156 DATA 50,70,90,100,110,126,140,176,200,210,220,240,260,280,400
158 DATA 50,70,92,100,140,150,172,182,200,210,230,240,260,280,400
160 DATA 50,70,90,100,120,130,140,150,174,184,200,210,230 240,260,280,400
162 DATA 50,70,90,100,120,130,140,150,176,186,204,236,260,280,400
164 DATA 50,70,90,100,120,130,140,150,178,188,206,274,260,280,400
166 RESTORE 148:FOR I=0 TO 170:READ P(I):NEXT I
168 I=0:TEST=0
170 REM TAROT
172 IF I=171 THEN 186
174 CALL LINE("G",P(I),L,P(I+1),L)
176 IF P(I+2)  400 THEN 180
178 RANDOMIZE:L=L+INTRND(4):I=I+3:GOTO 172
180 IF TEST=1 THEN 182 ELSE TEST=1:GOTO 172
182 TEST=0:I=I+2:GOTO 172
186 REM AFFICHAGE BOUTS
188 CALL COLOR("1BW"):LOCATE (14,10):PRINT "1  "
190 LOCATE (14,30):PRINT "21 "
192 CALL COLOR("1BW"):LOCATE (15,10):PRINT CHR$(1);CHR$(2);CHR$(3)
194 LOCATE (15,30):PRINT CHR$(1);CHR$(2);CHR$(3)
196 LOCATE (16,10):PRINT CHR$(4);CHR$(5);CHR$(6)
198 LOCATE (16,30):PRINT CHR$(4);CHR$(5);CHR$(6)
200 CALL COLOR("0BW"):LOCATE (17,10):PRINT "  1"
202 LOCATE (17,30):PRINT " 21"
204 CALL COLOR("1MW"):LOCATE (4,19):PRINT CHR$(42)
206 CALL COLOR("1MW"):LOCATE (4,20):PRINT CHR$(50);CHR$(51)
208 LOCATE (5,19):PRINT CHR$(52);CHR$(53);CHR$(54)
210 LOCATE (6,19):PRINT CHR$(55);CHR$(56)&" "
212 LOCATE (7,19):PRINT CHR$(57)
214 CALL COLOR("1MW"):LOCATE (7,20):PRINT " "&CHR$(42)
216 CALL COLOR("1MB"):LOCATE (20,1):PRINT "Voulez vous les regles ? 1=OUI 0=NON"
218 R=VAL(KEY$)
220 CLS:CALL HROFF
222 IF R=0 THEN 248
230 RESTORE 524:FOR I=0 TO 15:GOSUB 250:NEXT I
232 X$=KEY$:IF X$="" THEN 232
240 CLS:RESTORE 550:FOR I=0 TO 16:GOSUB 250:NEXT I
242 CALL COLOR("0RHL")
244 LOCATE (18,6):PRINT "BBOONNNNEE  CCHHAANNCCEE  !!!!"
245 LOCATE (19,6):PRINT "BBOONNNNEE  CCHHAANNCCEE  !!!!"
246 CALL COLOR("1YB"):LOCATE (22,6):PRINT "et entrez le second programme."
247 X$=KEY$:IF X$="" THEN 247
248 CLS:END                        247
249 REM AFFICHAGE
250 READ A$:L=0:N=LEN(A$)
252 IF A$="Tapez une touche pour sortir" THEN L=3
254 FOR Y=1 TO N
256 LOCATE (2+I+L,((40-N)/2)+Y)
258 PRINT SEG$(A$,Y,1)
260 NEXT Y
261 RETURN
382 REM BONJOUR
384 DATA 152,079,010,120,011,010,121,000,009,251,010
386 DATA 164,054,026,075,060,146,176,198,210,240,052,079,204,190,207,220,215
388 DATA 024,177,274,018,015,229,080,187,077,079,070,147,149,126,149,056,168
390 DATA 024,021,206,152,092,047,171,012,195,102,049,168,124,022,046,091,069
392 DATA 015,126,122,134,108,021,051,168,042,024,119,089,152,224,195,177,210
394 DATA 112,102,178,052,027,054,059,037,155,202,156,149,277,180,106,186,012
396 DATA 087,077,021,224,165,051,069,172,074,169,196,050,087,116,167,165,012
398 DATA 193,112,122,210,146,042,242,054,074,237,156,224,096,155,217,228,081
400 DATA 104,189,077,021,149,077,174,247,186,085,122,022,155,049,200,081,050
402 DATA 094,100,078,075,070,217,140,145,154,188,026,099,147,217,106,114,174
404 DATA 204,196,086,157,057,184,050,081,135,101,270,232,042,068,108,027,002
406 DATA 088,049,070,164,008,165,008,136,058,227,255,011
408 RESTORE 384:FOR I=0 TO 192:READ C
410 CALL POKE(50688+I,C):NEXT
412 CALL EXEC(50698)
414 PAUSE 1:RETURN
524 DATA "Les regles utilisees sont celles"
526 DATA "de la Federation Francaise de Tarot.
528 DATA "... ."
530 DATA "La donne est reglementaire,ainsi"
532 DATA "que les prises (petite,garde,sans-"
533 DATA "et contre le chien)"
534 DATA "L EXL 100 gere les jeux de est,nord"
536 DATA "et ouest independamment les uns"
538 DATA "des autres."
539 DATA "Il gere le decompte des points."
540 DATA "A vous de gagner ( ! ) avec le"
542 DATA "jeu de sud.Le choix d une carte se"
543 DATA "fait avec le manche (ou fleches)"
544 DATA "la validation du choix par TIR "
546 DATA "(ou espace)."
548 DATA "Tapez une touche pour sortir"
550 DATA "La manette orange suffit pour jouer."
552 DATA "Aucune frappe sur ENTER n est demandee."
554 DATA "... ...."
556 DATA "Les dernieres lignes sont reservees"
558 DATA "aux messages."
560 DATA "Lors du choix d une carte,si le curseur"
562 DATA "sort d un cote il reapparait de l autre."
564 DATA "Si vous avez une poignee a montrer,"
566 DATA "il faut taper successivement les atouts"
568 DATA "Ex: 15  tapez 1 puis 5 puis # pour"
570 DATA "valider,puis un autre atout."
572 DATA "A la fin apres le dernier #,tapez +"
574 DATA "Si vous montrez l excuse 1VEME,vous"
576 DATA "entrez 22."
578 DATA "Pour tous les autres cas il suffit de "
580 DATA "suivre les messages."
582 DATA "Tapez une touche pour sortir"
```